



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA**

**Máster Universitario en Informática Interactiva  
y Multimedia**

**Curso Académico 2012-13**

**Trabajo Final de Máster**

**Análisis de visualizaciones de la técnica de diseño de  
algoritmos de vuelta atrás**

**Autora: Natalia Esteban Sánchez**

**Tutor: Ángel Velázquez Iturbide**



# Resumen

En la enseñanza de la programación, en muchas ocasiones, se hace necesario apoyarse en una representación gráfica para poder explicar distintos conceptos de naturaleza abstracta. Una ilustración hace posible mostrar a los alumnos un modelo conceptual que les permita representar los conceptos mentalmente y les ayude a comprenderlos. Así, la visualización de algoritmos permite hacer aportes interesantes desde el punto de vista docente, permitiendo visualizar abstracciones de alto nivel que describen el algoritmo.

Existe numerosa bibliografía sobre algoritmia dedicada a la docencia que emplea representaciones gráficas para complementar e ilustrar las explicaciones textuales. Estas figuras permiten explicar conceptos de manera general o que tengan un carácter más específico sobre los algoritmos que se están tratando.

Si nos centramos en los criterios de representación, no se han encontrado criterios claros y explícitos sobre la creación de representaciones gráficas.

Es por ello por lo que, en este Trabajo Fin de Máster se presenta un análisis de las representaciones gráficas incluidas en libros de texto de prestigio sobre algoritmia con el objeto de poder establecer, posteriormente, las características que deben tener las visualizaciones docentes referentes a la técnica de vuelta atrás, para deducir los principios de diseño gráfico de las visualizaciones e implementar un sistema de visualización.



## Agradecimientos

En primer lugar, agradecer a mi tutor Ángel todas las correcciones, apoyo y seguimiento de este trabajo y por darme la oportunidad de poder participar en un proyecto tan interesante. Me has enseñado mucho y me has dado una visión diferente que no conocía sobre la participación en proyectos y el trabajo en equipo. ¡Muchas gracias!

Un rincón muy especial de esta hoja va dedicado a mi familia. A Cuel, por todo tu cariño, comprensión, tus ánimos, por saber animarme cuando más lo he necesitado y por hacer todo mucho más fácil. A mi madre, Mari, que en todo momento me has animado a conseguir lo que me proponga y me has enseñado a ser constante y luchadora. Muchas gracias por inculcarme grandes valores, por tu ejemplo de fuerza y coraje, por tu optimismo y por tu amor. Como no, muchas gracias a Paula, que todavía no estás aquí con nosotros y ya me estás dando alegrías e ilusiones. Me das la paz y tranquilidad que siempre he necesitado.

A Silvy, por ser tan especial como eres y por escucharme y apoyarme en todo momento. Tengo mucha suerte de tenerte a mi lado. ¡No cambies nunca!

Gracias a Tamara, a Julio y a ese “nuevo” descubrimiento, Sara, que has estado un año a mi lado y has pasado desapercibida, me alegro de conocerte y poder contar con una persona como tú. Gracias a los tres por vuestro apoyo y comprensión.



# Índice general

<b>Resumen .....</b>	<b>II</b>
<b>Agradecimientos .....</b>	<b>IV</b>
<b>Índice general .....</b>	<b>VI</b>
<b>Índice de figuras .....</b>	<b>VII</b>
<b>Índice de tablas .....</b>	<b>VIII</b>
<b>1. Motivación .....</b>	<b>1</b>
1.1 <i>Presentación del problema .....</i>	3
1.2 <i>Objetivos del trabajo .....</i>	4
1.3 <i>Estructura del documento.....</i>	4
<b>2. Recogida y análisis de figuras .....</b>	<b>7</b>
2.1 <i>Recogida: método y estadísticas.....</i>	7
2.2 <i>Análisis: etapas e iteraciones realizadas .....</i>	11
2.2.1 <i>Categorías de catalogación preliminares .....</i>	18
2.2.2 <i>Resultado del análisis preliminar .....</i>	19
<b>3. Categorías de catalogación .....</b>	<b>23</b>
3.1 <i>Objetivo de representación .....</i>	25
3.2 <i>Genericidad .....</i>	33
3.3 <i>Representación y completitud del estado.....</i>	34
3.4 <i>Representación y completitud del tiempo .....</i>	36
<b>4. Análisis de figuras .....</b>	<b>39</b>
4.1 <i>Resultado del análisis de figuras .....</i>	52
<b>5. Conclusiones y trabajos futuros.....</b>	<b>57</b>
5.1 <i>Conclusiones.....</i>	57
5.2 <i>Trabajos futuros .....</i>	58
<b>Bibliografía .....</b>	<b>61</b>
<b>Anexo I – Problema planteado a los alumnos .....</b>	<b>65</b>

## Índice de figuras

Figura 1. Árbol de búsqueda abreviado potencial.....	14
Figura 2. Diagrama etapas e iteraciones realizadas .....	18
Figura 3. Tablero movimientos del caballo de ajedrez .....	26
Figura 4. Grafo planar-grafo .....	27
Figura 5. Tablero de ajedrez 8-reinas .....	27
Figura 6. Tablero de ajedrez 8-reinas .....	28
Figura 7. Ejemplo de partida del dominó.....	28
Figura 8. Tablero de ajedrez - comprobación de validez .....	29
Figura 9. Árbol de búsqueda potencial .....	29
Figura 10. Tablero parcial con movimientos del caballo válidos.....	30
Figura 11. Numeración por niveles – cuadrado latino.....	30
Figura 12. Árbol de búsqueda generado – tablero solución .....	31
Figura 13. Tablero 4-reinas – 2 estados .....	31
Figura 14. Árbol de búsqueda generado.....	32
Figura 15. Árbol de búsqueda generado.....	33
Figura 16. Árbol de búsqueda abreviado .....	33
Figura 17. Árbol de búsqueda potencial .....	34
Figura 18. Grafo completo .....	35
Figura 19. Árbol parcial generado .....	37
Figura 20. Tableros – secuencia + solapamiento.....	38



## Índice de tablas

Tabla 1. Tabla resumen por autor – información recopilada.....	9
Tabla 2. Tabla resumen tipo de problema– información recopilada .....	9
Tabla 3. Porcentajes de figuras por categoría de problemas.....	11
Tabla 4. Tabla Excel – Categorías de catalogación definitivas .....	25
Tabla 5. Subobjetivos de representación por objetivos de representación .....	25
Tabla 6. Porcentajes de figuras por genericidad .....	39
Tabla 7. Porcentajes de figuras por objetivo de representación .....	39
Tabla 8. Porcentajes de figuras por subobjetivo de representación .....	40
Tabla 9. Porcentajes de figuras por representación del estado.....	40
Tabla 10. Porcentajes de figuras por completitud del estado .....	40
Tabla 11. Porcentajes de figuras por representación del tiempo .....	40
Tabla 12. Porcentajes de figuras por completitud del tiempo para árboles de búsqueda .....	41
Tabla 13. Porcentajes de figuras por atributos de representación .....	41
Tabla 14. Porcentajes de figuras por clase de problemas y autores .....	41
Tabla 15. Porcentajes de figuras por clase de problemas y genericidad.....	42
Tabla 16. Porcentajes de figuras por clase de problemas y representación del estado .....	42
Tabla 17. Porcentajes de figuras por clase de problemas y representación del tiempo .....	43
Tabla 18. Porcentajes de figuras por representación del espacio y representación del tiempo .....	43
Tabla 19. Porcentajes de figuras por atributos de representación y autor .....	44
Tabla 20. Porcentajes de figuras por atributos de representación y completitud del tiempo .....	44
Tabla 21. Tipos de árbol abreviados .....	44
Tabla 22. Porcentajes de figuras por completitud del tiempo para árboles de búsqueda y autores.....	45
Tabla 23. Porcentajes de figuras por objetivo de representación y clase de problemas .....	45
Tabla 24. Porcentajes de figuras por subobjetivo de representación y clase de problemas.....	46
Tabla 25. Porcentajes de figuras por objetivo de representación y autores .....	46
Tabla 26. Porcentajes de figuras por subobjetivo de representación y autores.....	47
Tabla 27. Porcentajes de figuras por subobjetivo de representación y genericidad ....	47
Tabla 28. Porcentajes de figuras por objetivo de representación y representación del estado.....	48
Tabla 29. Figuras por subobjetivo de representación y representación del estado ....	48
Tabla 30. Porcentaje de figuras por objetivo de representación y representación del tiempo .....	49
Tabla 31. Porcentaje de figuras por objetivo y atributos de representación .....	49
Tabla 32. Porcentaje de formatos en árboles y clases de problemas .....	50
Tabla 33. Porcentajes tipo etiquetas nodos de árboles y representación simbólica ....	50

Tabla 34. Porcentajes distinción nodos y representación simbólica .....	51
Tabla 35. Porcentajes de formatos en tableros y categoría juegos (J) .....	51
Tabla 36. Porcentajes de formatos en tableros y categoría grafos (G) .....	51
Tabla 37. Porcentajes de formatos de árboles y autores .....	52

# 1. Motivación

La enseñanza de materias referentes a la programación es un reto para los profesores universitarios de informática debido a que muchos conceptos de programación, como por ejemplo la recursividad, son abstractos por naturaleza y otros no tienen una contrapartida en el mundo real que le permita al alumno establecer un modelo mental que le ayude a comprenderlo. Por ello, informáticos, pedagogos y psicólogos han dedicado grandes esfuerzos a mejorar su enseñanza desde distintos enfoques. Uno de estos enfoques se centra en el uso de visualizaciones de algoritmos [1]. La visualización de algoritmos consiste en la visualización de abstracciones de alto nivel que describen el algoritmo de modo que a los alumnos les sea más fácil entenderlos.

Un hecho sorprendente es que, durante los últimos años, las aportaciones al campo de la visualización de algoritmos se han centrado en cuestiones pedagógicas provocando cierta falta de interés por las propias visualizaciones. Sin embargo, no podemos ignorar su importancia. En muchos casos, la naturaleza dinámica de los algoritmos hacen que sean difíciles de explicar sin utilizar ilustraciones que representen su comportamiento. Así, cuando se utiliza una visualización de un algoritmo, sea producida por un sistema software, incluida como ilustración de un libro o dibujada por el docente en la pizarra, se está mostrando un modelo conceptual a los alumnos para transmitirles el comportamiento del algoritmo en ejecución.

La visualización de algoritmos se puede considerar transversalmente como una representación estática y dinámica. La representación estática está basada en la esquematización de la estructura de su especificación y la representación dinámica, que se conoce como animación de algoritmos, muestra su comportamiento en tiempo de ejecución.

Hay multitud de investigaciones sobre recomendaciones referentes a sistemas de visualización de algoritmos, relacionadas con la interfaz de usuario, las propias animaciones y distintas funciones aconsejables para la adopción de sistemas. Así, como se deduce de [2], sería aconsejable que los sistemas cumplieran distintos recursos pedagógicos como por ejemplo:

- Ser sistemas de propósito general y no aplicarse a aspectos muy concretos.
- Permitir que el alumno pueda introducir datos de entrada.
- Deben incluir preguntas predictivas que les permita disponer de predicción interactiva.
- ...

Sin embargo, tras examinar la literatura académica, no se deduce una lista clara de características que permitan definir las propiedades que debe cumplir un sistema. Se trata de un tema de difícil estudio ya que no sólo intervienen las características de las visualizaciones sino otros factores tales como la tarea docente a realizar y la implicación por parte del alumnado.

Si nos centramos en las recomendaciones sobre las propias visualizaciones, encontramos algunas sugerencias sobre técnicas de visualización de la información o técnicas básicas de comunicación visual pero no encontramos reglas de uso.

En otros casos encontramos recomendaciones sobre el diseño de representaciones gráficas para algoritmos pero son demasiado generales o específicas. Sami Khuri [3] presenta un enfoque basado en el usuario y comenta las fases de análisis y de diseño del sistema. Contiene recomendaciones interesantes sobre la interfaz de usuario (diseño de la pantalla, múltiples vistas y formas de interacción) y sobre algunas técnicas visuales (color, sonido y 3D).

En cuanto al formato gráfico de las visualizaciones, sólo encontramos propuestas claras para algoritmos recursivos. Es el caso de Stern y Naish [4] que utilizan las animaciones de algoritmos como herramienta pedagógica, presentan una clasificación de los algoritmos recursivos en tres clases (algoritmos que trabajan con estructuras de datos estáticas, algoritmos que modifican las estructuras de datos y los algoritmos que construyen una estructura de datos). Partiendo de esta clasificación presentan visualizaciones que consideran adecuadas para cada una de las clases. Asimismo, Velázquez et al. [5] proponen y justifican tres representaciones gráficas para algoritmos de divide y vencerás.

Existe numerosa bibliografía sobre algoritmia que se apoya en representaciones

gráficas para completar e ilustrar distintas explicaciones textuales y aportar ejemplos que ilustren conceptos de algoritmos específicos. Pero no hemos encontrado criterios claros y explícitos sobre la creación de representaciones gráficas. Pero existe experiencia sobre cómo llevar a cabo una investigación encaminada a encontrarlos.

## 1.1 Presentación del problema

Los “principios de diseño” son reglas de alto nivel que describen características de las visualizaciones más eficaces en un dominio. En algunos dominios, como por ejemplo en el campo de los grafos o la estadística, existen estos principios de diseño, pero en otros no.

Así, Maneesh Agrawala [6] presenta varios trabajos para aplicaciones tan dispares como mapas y rutas, mapas turísticos, instrucciones de ensamblaje, e ilustraciones basadas en cortes o expansiones para entidades mecánicas, matemáticas o biológicas. Agrawala, contempla distintas estrategias a seguir, según la madurez del dominio: análisis de visualizaciones manuales, uso de trabajos previos en percepción y cognición, o realización de experimentos de percepción y cognición.

Teniendo en cuenta que existe numerosa bibliografía sobre algoritmia que se apoya en representaciones gráficas “manuales” para completar e ilustrar distintas explicaciones textuales y aportar ejemplos que ilustren conceptos específicos y siguiendo el método de trabajo de Agrawala [6], se ha emprendido un proyecto de análisis de dichas visualizaciones. Concretamente, el proyecto se centra en analizar las visualizaciones propias de algunas de las técnicas de diseño de algoritmos más comunes: divide y vencerás, vuelta atrás y programación dinámica.

En concreto, se han analizado las ilustraciones contenidas en libros de texto de prestigio sobre algoritmia con el objeto de poder establecer las características que deben tener las visualizaciones docentes de técnicas de diseño de algoritmos. Estas características serán la base para la especificación de principios de diseño de visualizaciones de algoritmos que a su vez serán la base para la implementación de una herramienta software para la visualización de algoritmos.

Este Trabajo Fin de Máster se centra en analizar las visualizaciones existentes en los libros de texto de prestigio sobre la técnica de diseño de vuelta atrás para poder establecer así las características que deben tener las visualizaciones docentes sobre dicha técnica y posteriormente poder establecer los principios de diseño de visualizaciones.

## 1.2 Objetivos del trabajo

Partiendo de la hipótesis de que las representaciones empleadas en los libros de texto de prestigio sobre algoritmia son las más extendidas, aceptadas y útiles en el contexto docente, el objetivo principal del Trabajo Fin de Máster es realizar un análisis de dichas representaciones que permita extraer las características que deben tener las visualizaciones docentes sobre la técnica de diseño de vuelta atrás y poder establecer así, posteriormente, los principios de diseño de visualizaciones de algoritmos referidos a dicha técnica.

Para alcanzar el objetivo principal, se ha fijado el siguiente plan de trabajo:

- Fase 1: Realizar una recopilación de los problemas utilizados en los libros de texto para ilustrar la técnica de diseño de vuelta atrás. Dicha recopilación debe contener la siguiente información:
  - Las figuras o ilustraciones que se incluyen en los diferentes problemas planteados por cada autor.
  - El código o pseudocódigo que resuelve el problema planteado.
- Fase 2: Analizar las figuras recogidas, categorizar los tipos de problemas y establecer las categorías de catalogación necesarias para englobar todas las características y propiedades que presentan las imágenes y poder realizar así el análisis estadístico de dichas figuras.
- Fase 3: Llevar a cabo el análisis estadístico y presentar los resultados.

## 1.3 Estructura del documento

La organización de los estantes capítulos de esta memoria es la siguiente:

**Capítulo segundo:** Se incluye una descripción del método llevado a cabo para la

recogida de figuras sobre la técnica de vuelta atrás, datos estadísticos de número de libros y figuras, un análisis de las fases e iteraciones realizadas para poder realizar un análisis estadístico preliminar y un breve resumen estadístico de dicho análisis preliminar.

**Capítulo tercero:** Se presenta una descripción de las categorías establecidas para realizar el análisis de figuras, incluyendo una sección por cada categoría principal.

**Capítulo cuarto:** Se especifica el análisis detallado sobre las características de las figuras de algoritmos presentes en los libros de texto para la técnica de vuelta atrás.

**Capítulo quinto:** Contiene las conclusiones obtenidas tras finalizar el trabajo, y se introducen posibles trabajos futuros que se podrían abordar como ampliación de este Trabajo Fin de Máster.





## 2. Recogida y análisis de figuras

A continuación se expone el método llevado a cabo para la recogida de figuras, así como el análisis de las fases e iteraciones realizadas para poder elaborar el análisis estadístico de las ilustraciones incluidas en los libros de texto.

### 2.1 Recogida: método y estadísticas

#### Método para la recogida de figuras

El trabajo realizado en este Trabajo fin de Máster comenzó bajo la hipótesis de que las representaciones empleadas en los libros de texto de prestigio sobre algoritmia son las más extendidas, aceptadas y útiles en el contexto docente. Por ello, se decidió iniciar una exploración de la bibliografía existente sobre algoritmia disponible en la biblioteca universitaria, seleccionando únicamente aquellas obras centradas en el diseño de algoritmos. En concreto, para llevar a cabo la recogida, en cada libro, en primer lugar se ha buscado si en el índice había algún capítulo sobre vuelta atrás o algoritmos de búsqueda en espacio de estados. En caso afirmativo, se ha realizado la recogida en el mismo; si no, se ha buscado en el glosario de términos el nombre de problemas conocidos como por ejemplo, el problema de las  $n$ -reinas, del salto de caballo, del laberinto, etc. Lo que se pretendía recolectar era cualquier representación gráfica que fuese utilizada para ilustrar un concepto o algoritmo, bien de manera parcial o total.

Para realizar la recopilación se han tomado como base documental un total de 14 obras [7][8][9][10][11][12][13][14][15][16][17][18][19][20], recopilando información sobre las características de cada representación y estableciendo una relación entre cada problema, qué representaciones tenía disponibles y el código o pseudocódigo que explique los diferentes pasos a seguir para la resolución de un problema.

En un primer lugar, se ha llevado a cabo una revisión bibliográfica [21] organizando los problemas en los siguientes grupos:

1. Problemas de decisión
  - 1.1. Problema de las 8-reinas

- 1.2. Problemas de grafos
- 1.3. Problemas de cadenas de caracteres
- 1.4. Problema del salto de caballo
- 1.5. Problema del laberinto
- 1.6. Problemas de juegos
  - 1.6.1. Problema del cuadrado mágico
  - 1.6.2. Problema del latino
  - 1.6.3. Problema del dominó
  - 1.6.4. Problema del sudoku
- 1.7. Problemas de subconjuntos
- 1.8. Otros problemas de decisión
  - 1.8.1. Problema de las factorías
  - 1.8.2. Problema de franquear postales
  - 1.8.3. Problema del reparto
- 2. Problemas de optimización
  - 2.1. Problema de la mochila 0/1
  - 2.2. Problemas de asignación
    - 2.2.1. Asignación de tareas
  - 2.3. Otros problemas de optimización
    - 2.3.1. Problema de recolección
    - 2.3.2. Problema de los comensales
    - 2.3.3. Problema de cableado de longitud mínima
    - 2.3.4. Problema de franquear postales
    - 2.3.5. Problema de minimizar el número de envases

En esta revisión bibliográfica, con el fin de mantener la originalidad del problema que aparece en cada obra, se han recogido dichos problemas de forma literal, es decir, mantenido el idioma y el nombre.

Además, para cada problema, se ha incluido el enunciado general del mismo y el enunciado encontrado en cada uno de los libros revisados, así como aquellas figuras y código o pseudocódigo utilizado por los autores en cada problema. Se han escaneado tanto las figuras como el código o pseudocódigo incluido en cada una de las obras.

Así, con el objeto de proporcionar el máximo de información resumida sobre cada problema incluido en cada una de las obras revisadas, se confeccionó unas tablas de resumen con el formato mostrado en Tabla 1.

Libro	Capítulo/apartado	Visualización	Implementación

Tabla 1. Tabla resumen por autor – información recopilada

De este modo se recoge el capítulo y/o apartado donde se encuentra la visualización, el número de página y tipo de ilustración y si el autor incluye la implementación del algoritmo. El hecho de incluir esta información facilita la lectura y ayuda a poder localizar de forma rápida la ilustración que se quiere consultar.

Al finalizar la recopilación de cada tipo de problema, con el objeto de facilitar la búsqueda e identificación de qué obras incluyen figuras, también se ha incluido una tabla de resumen (véase Tabla 2) con la información que se puede encontrar en cada uno de los libros. En este caso se ha incluido también la nomenclatura utilizada por cada autor para el problema planteado.

Libro	Capítulo/apartado	Visualización	Nomenclatura	Implementación

Tabla 2. Tabla resumen tipo de problema– información recopilada

Tras revisar toda la bibliografía y realizar una primera clasificación de las figuras incluidas en las distintas obras, con el objeto de poder identificar distintas tendencias de representación entre los distintos autores, se han reagrupado los tipos de problemas. Así, el análisis definitivo (véase capítulo 4) de las figuras correspondientes a los distintos tipos de problemas se ha llevado a cabo atendiendo a la siguiente clasificación:

- 1) Problemas de juegos (J): Los problemas englobados dentro de esta categoría son: el laberinto, la ubicación de 8 reinas en un tablero de ajedrez, el salto de caballo, el cuadrado mágico y cuadrado latino, el dominó y el sudoku.
- 2) Problemas sobre grafos (G): Esta categoría contiene los siguientes problemas:

coloreado de grafos, encontrar todos los ciclos hamiltonianos de un grafo, construir todos los caminos de un grafo, grafos isomorfos.

- 3) Problemas de cadenas de caracteres (C): En esta categoría se incluyen los problemas: generación de palabras con restricciones, variaciones y permutaciones.
- 4) Problemas de decisión (D): Donde se incluye el problema de calcular los subconjuntos, el problema de las factorías, el problema del franqueo de postales y el problema del reparto.
- 5) Problemas de optimización (O): Los problemas contenidos dentro de esta categoría son: mochila 0/1, asignación de tareas, recolección, cableado de longitud mínima, comensales, franquear postales y minimizar el número de envases.

### **Datos estadísticos**

En la recopilación de figuras se recogen 25 tipos de problemas de entre las 14 obras bibliográficas consultadas para la realización del informe. De los 25 tipos de problemas 6 no cuentan con ninguna representación gráfica. Se han recopilado un total de 55 ilustraciones, si bien algunas son dobles, por lo que a efectos estadísticos existen un total de 62. La media de representación por problema es de 2.48.

De los autores consultados, seis [9][12][15][16][19][20] no incluyen la técnica de vuelta atrás en ninguno de sus capítulos.

En la Tabla 3 se muestra el porcentaje de las representaciones halladas según la clasificación a la que pertenecen. Puede observarse que la categoría con mayor número de representaciones es la de juegos y la categoría con menor número de representaciones es la de cadenas. Esto tiene sentido ya que los problemas de juegos suelen incluir figuras para probar distintos movimientos posibles por parte del jugador.

Total	J	G	C	D	O
62(100%)	25(40,32%)	18(29,03%)	1(1,61%)	10(16,13%)	8(12,90%)

Tabla 3. Porcentajes de figuras por categoría de problemas

## 2.2 Análisis: etapas e iteraciones realizadas

El Trabajo Fin de Máster que se presenta forma parte de un proyecto de investigación en el que se analizan tres técnicas de diseño de algoritmos: divide y vencerás, vuelta atrás y programación dinámica. En la realización del análisis han participado 4 investigadores, uno por cada técnica de diseño analizada y un cuarto investigador que coordina y supervisa los análisis individuales.

El análisis de las ilustraciones se ha llevado a cabo en distintas etapas, y en ocasiones, en cada etapa, con el objeto de unificar y homogeneizar criterios para las tres técnicas de diseño, se han realizado varias iteraciones de trabajo grupal. Así, podemos diferenciar las siguientes etapas:

### **Etapas 1: recogida de figuras y realización de informe técnico**

En esta primera etapa, cada uno de los investigadores realizó independientemente su recogida de ilustraciones. Para la recogida, se han seleccionado libros de prestigio escritos en inglés, pero también nos ha parecido adecuado analizar libros en español. En principio, no se ha cuestionado la idoneidad de los algoritmos incluidos por cada autor en cada una de las técnicas de diseño.

Cada investigador escaneó las ilustraciones encontradas y elaboró un informe técnico: vuelta atrás [21], divide y vencerás [22] y programación dinámica [23]. Cada informe agrupa los problemas en categorías (véase apartado 2.1 para la técnica de vuelta atrás); dada la especificidad de cada técnica de diseño, en esta primera etapa, las categorías fueron distintas. Para cada problema, se reprodujo literalmente su enunciado y sus ilustraciones y, con el objeto de poder acceder fácilmente a la información, se recopiló la información bibliográfica asociada (por ejemplo, capítulo y página), nomenclatura del problema e información sobre las ilustraciones y su código fuente.

Cabe destacar que, en primer lugar, una persona realizó la recopilación de ilustraciones para la técnica de divide y vencerás y meses más tarde otra persona realizó la recopilación para la técnica de programación dinámica y una tercera persona llevó a cabo la recopilación para la técnica de vuelta atrás utilizada para el análisis presentado en este proyecto. Estas dos últimas personas han realizado la recopilación más o menos en paralelo. Esta separación temporal ha influido en la lista de libros analizados. Al final, se han analizado un total de 21 libros, de los que sólo 8 son comunes a las tres técnicas de diseño. Siete libros se han analizado solamente para la técnica de divide y vencerás, y 6 sólo para vuelta atrás y programación dinámica.

### **Etapas 2: revisión de los informes, conclusiones preliminares y primer análisis**

Una vez realizados los informes técnicos para cada una de las técnicas de diseño, una cuarta persona los leyó y realizó un análisis exploratorio. Como resultado, elaboró un documento con características destacables, y elaboró una serie de características interesantes que precisaba un análisis más detallado.

Las conclusiones preliminares que se extrajeron son las siguientes:

- En todas las técnicas de diseño de algoritmos se utilizaban tanto representaciones gráficas de pendientes del dominio (por ejemplo, un diagrama geométrico) como propias de la informática (por ejemplo, la pila de control).
- Dado un problema o algoritmo, las ilustraciones se utilizan con diversos objetivos, sobre todo como ejemplo, para una ejecución o para una explicación. En los dos primeros casos se usan datos concretos, mientras que en el último se plantea de forma genérica.
- Las figuras no sólo incluyen valores, sino también identificadores para aumentar su comprensión.
- Presentación frecuente de una secuencia de estados de la ejecución de un algoritmo.
- Algunas ilustraciones sugieren operaciones de interacción en un sistema de visualización de software, por ejemplo, conexión entre elementos.
- Algunas figuras aparecen simplificadas de forma análoga a la técnica de

foco+contexto.

Posteriormente nos reunimos en varias ocasiones para comentar las conclusiones extraídas y para determinar los pasos a seguir en las siguientes etapas. Además, se establecieron un conjunto inicial de categorías de catalogación y se definieron definiendo las características de generalidad y completitud de las ilustraciones.

Tras esta primera revisión, cada investigador encargado de cada técnica de diseño llevó a cabo un primer análisis. Para ello, para la técnica de vuelta atrás, se revisaron las categorías de problemas definidas, reduciendo su número, y se analizaron las relaciones entre las representaciones gráficas, clases de problemas y objetivos de las figuras.

Para la técnica de vuelta atrás se establecieron las siguientes categorías de problemas:

- 1) Problemas de juegos y estrategia (JE): Los problemas englobados dentro de esta categoría son: el laberinto, la ubicación de 8 reinas en un tablero de ajedrez, el salto de caballo, el cuadrado mágico y cuadrado latino, el dominó y el sudoku.
- 2) Problemas sobre grafos (G): Esta categoría contiene los siguientes problemas: coloreado de grafos, encontrar todos los ciclos hamiltonianos de un grafo, construir todos los caminos de un grafo, grafos isomorfos.
- 3) Otros problemas de decisión (OD): Donde se incluye el problema de calcular los subconjuntos, el problema de las factorías, el problema del franqueo de postales y el problema del reparto.
- 4) Problemas de optimización (PO): Los problemas contenidos dentro de esta categoría son: mochila 0/1, asignación de tareas, recolección, cableado de longitud mínima, comensales, franquear postales y minimizar el número de envases.

Para realizar el análisis estadístico para la técnica de vuelta atrás se elaboró una tabla Excel donde se recogía, a modo de columna, para cada figura la siguiente información:

- Autor, nomenclatura del problema, y el estilo de las figuras (si incluyen etiquetas, el formato de los nodos, como representa los elementos del tablero, para los árboles de búsqueda, si tiene etiquetas en el nodo, dentro o fuera del mismo, el tipo de etiqueta, si tiene etiquetas en las ramas, etc.).
- Clase de representación: en las figuras incluidas en vuelta atrás encontramos, árboles de búsqueda, figuras dependientes del dominio (por ejemplo, tablero, mapa), representaciones propias de la informática (tablas, matriz, grafos).
- Objetivo de representación: se distinguen los siguientes objetivos: explicación, proceso de cómputo, entrada, solución, entrada/salida.
- Tipo de árbol: Los árboles de búsqueda puede ser potenciales o generados. Además, los árboles de búsqueda se pueden representar de forma abreviada. Se distinguen entre tres tipos de árboles abreviados: árboles con subárboles suprimidos (SS, véase Figura 9), árboles con subárboles comprimidos en triángulo o suprimidos, y arcos suprimidos con puntos suspensivos (CTAPS, véase Figura 16), y con subárboles comprimidos en triángulo (ST, véase Figura 1).

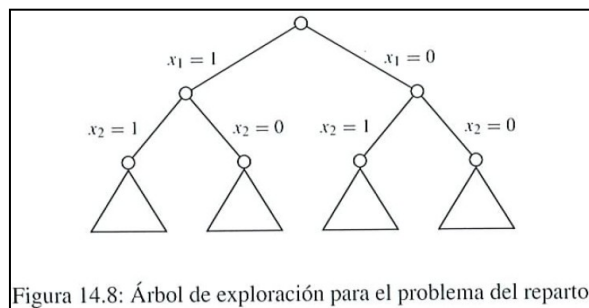


Figura 1. Árbol de búsqueda abreviado potencial

### **Eta­pa 3: puesta en comu­n, categor­as y an­lisis preliminar**

Una vez elaborados los an­lisis estad­sticos para cada t­cnica de dise­no, los resultados se pusieron en comu­n, identificando tendencias comunes. Esta puesta en comu­n nos permiti­ identificar los aspectos que faltaban por analizar para cada t­cnica. Por ejemplo, para la t­cnica de divide y vencer­s se vio la necesidad de clasificar los algoritmos encontrados.

Tras establecer las categor­as que deb­amos utilizar para las t­nicas de dise­no, cada



investigador actualizó su tabla Excel y volvió rehacer su análisis en función de las nuevas categorías.

Después nos volvimos a juntar con el investigador principal para poner los resultados en común y nos dimos cuenta de que cada investigador había considerado propiedades y atributos distintas según la técnica de diseño y al poner los análisis en común había propiedades que no se podían clasificar en ninguna de las categorías que habíamos definido. Esto dificultaba el hecho de poder llevar a cabo un análisis homogéneo para las tres técnicas con el fin de establecer principios de diseño comunes de las visualizaciones.

De nuevo el investigador principal, que había mantenido distintas reuniones con el resto de investigadores por separado y conocía las imágenes recopiladas para las tres técnicas de diseño, elaboró un guión para llevar a cabo un nuevo análisis donde todos trabajásemos sobre los mismos campos y los resultados fuesen homogéneos para las tres técnicas de diseño.

Una vez elaborado el guión, mantuvimos cada persona encargada de cada técnica de diseño con el investigador principal y repasamos las imágenes y la clasificación de las nuevas categorías (véase apartado 2.2.1).

Cada investigador encargado de cada una de las técnicas de diseño rehízo el Excel ajustándolo a las nuevas categorías (véase apartado 2.2.1) y realizó un análisis estadístico preliminar. Este análisis estadístico preliminar ha sido presentado en [25] y [26].

### **Etapas 4: categorías de catalogación definitivas y análisis estadístico**

Tras realizar el análisis preliminar había propiedades de las figuras que no cuadraban del todo con las categorías establecidas hasta el momento y por ello, investigador principal rehízo el guión para el análisis de figuras, modificando nuevamente las categorías de catalogación de modo que se pudiesen englobar en ellas de una forma lo más homogénea posible todas las propiedades de todos los algoritmos. Además, también se redefinieron las clasificaciones de los problemas. Para la técnica de vuelta

atrás se establecieron las categorías de problemas indicadas en el apartado 2.1 (inicialmente se partió de la catalogación de problemas mostrada en la etapa 2).

En esta ocasión se definió un Excel común con las columnas que deberíamos ir completando con las propiedades de cada figura correspondiente a cada técnica de diseño.

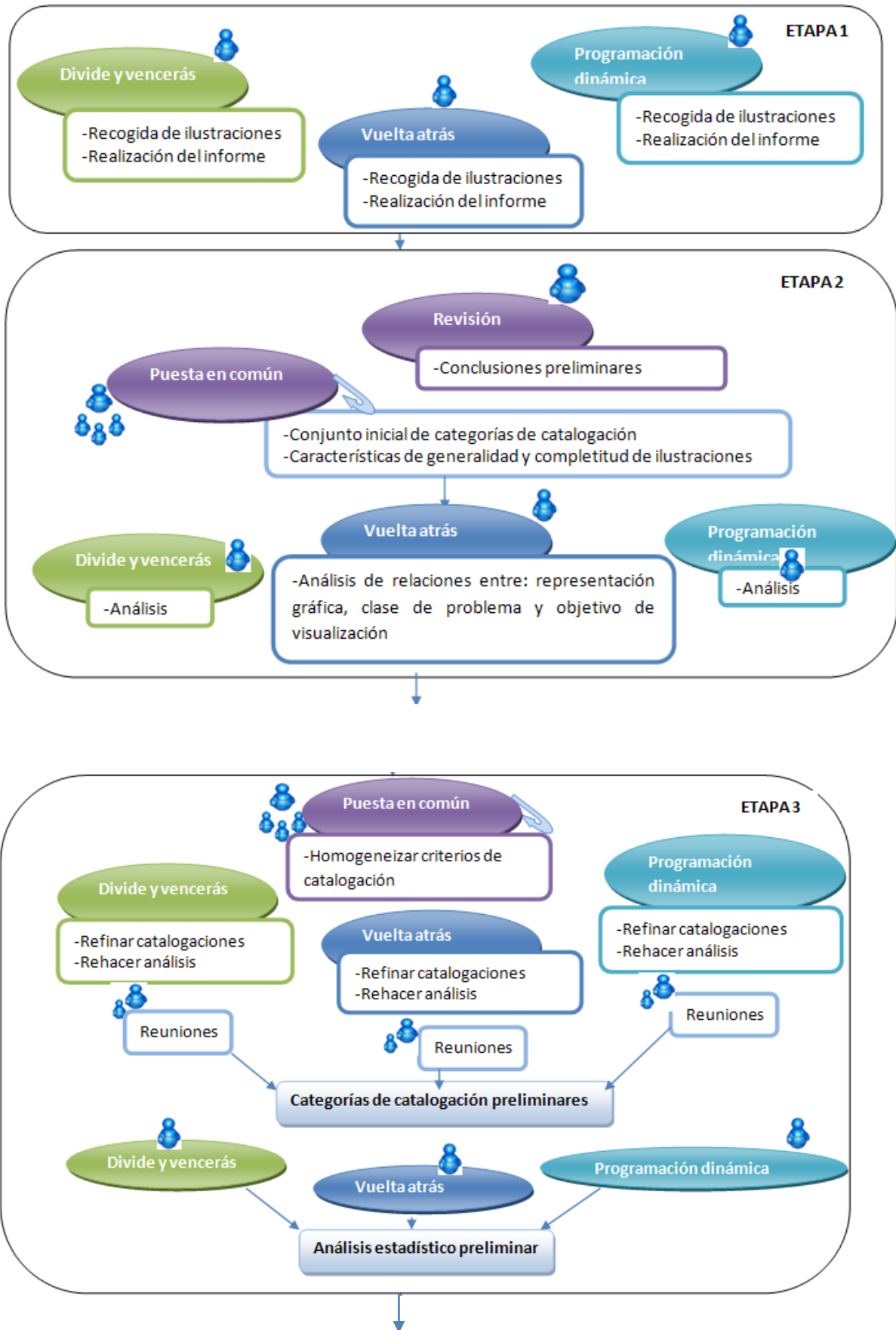
Posteriormente se realizaron varias rondas en las que se refinaron los criterios de categorización y se homogeneizaron los resultados de los tres análisis. Para aumentar la fiabilidad de la clasificación, el cuarto investigador se reunió por separado con cada uno de los tres investigadores que habían recopilado las ilustraciones y revisaron juntos las catalogaciones y revisaron el Excel generado revisando una a una cada una de las figuras recopiladas para cada técnica de diseño. Las categorías de catalogación resultantes se muestran en el capítulo 3.

Para la técnica de vuelta atrás cambia también el número de figuras simples, al dividir algunas figuras en varias figuras simples. Para el análisis preliminar se consideraron 60 ilustraciones simples, mientras que para el análisis final se cuenta con un total de 64.

Finalmente cada investigador por separado ha realizado el análisis estadístico para cada una de las técnicas de diseño.

El análisis estadístico para la técnica de vuelta atrás se incluye en el apartado 4.

En la Figura 2 se muestra un resumen de las etapas llevadas a cabo en el proceso.



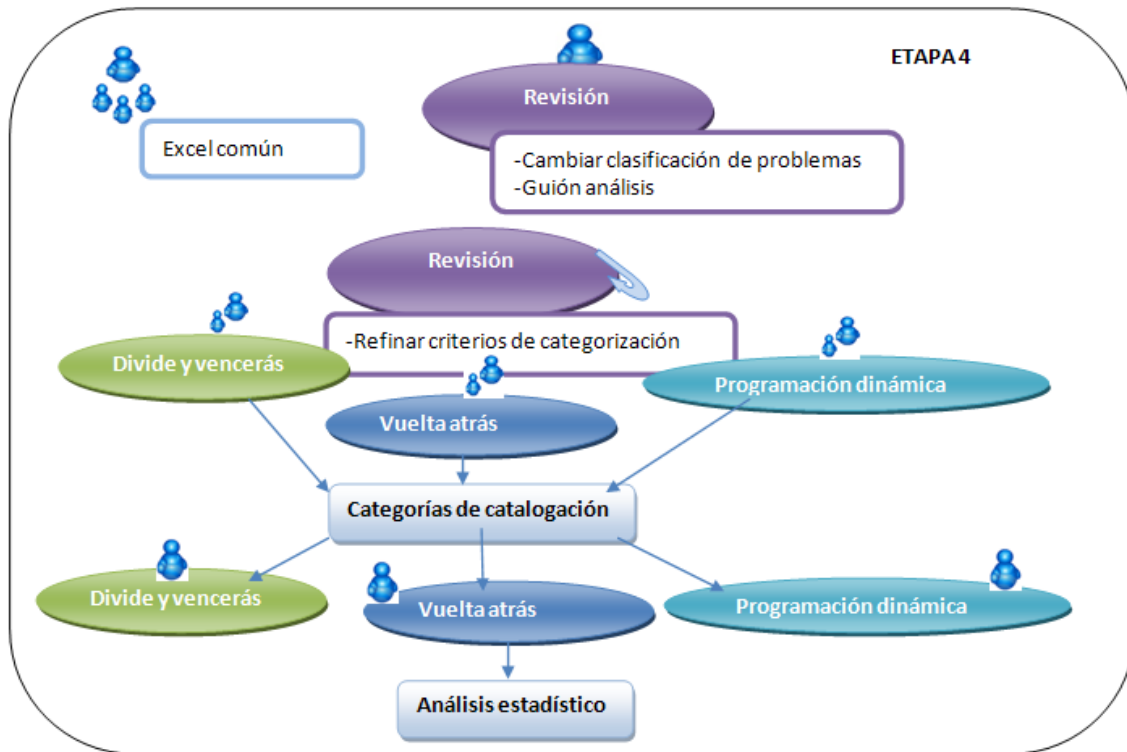


Figura 2. Diagrama etapas e iteraciones realizadas

### 2.2.1 Categorías de catalogación preliminares

Inicialmente no disponíamos de categorías para clasificar las ilustraciones, sino que se refinaron al avanzar el análisis, siguiendo los principios de la *grounded theory* [24].

Una primera dificultad con la que nos encontramos es la definición de ilustración, que es necesaria para su contabilidad. Consideramos que una ilustración es lo que un autor ha considerado como tal, salvo que no sea relevante para nuestro análisis (por ejemplo, si la figura contiene código fuente). Sin embargo, al realizar el análisis encontramos numerosas ilustraciones compuestas y, a la inversa, ilustraciones consecutivas muy parecidas. Resolvimos esta situación distinguiendo entre los conceptos ilustración e ilustración simple. Si una ilustración consta de varias con distinto formato u objetivo (véase Figura 12), se parte en varias ilustraciones simples. Asimismo, si varias ilustraciones consecutivas son muy parecidas, con el mismo formato y objetivo (por ejemplo, una secuencia de estados de ejecución), se unen en una sola ilustración simple (véase Figura 13).

Las categorías que hemos establecido para realizar el análisis preliminar son:

- Clase de problema. Nuestra aproximación ha sido pragmática, identificando clases de problemas que agrupan de forma “razonable” los problemas encontrados para cada técnica de diseño. Por ejemplo, problemas de grafos, de juegos o matemáticos.
- Representación gráfica. Distinguimos todas las clases de representaciones gráficas necesarias (por ejemplo, vector, tablero, etc.). Algunas son parecidas pero se diferencian en su diseño gráfico (por ejemplo tabla y matriz). Una segunda clasificación de las representaciones gráficas distingue entre representaciones dependientes del dominio (por ejemplo, tablero), representaciones propias de la informática (por ejemplo, vector) y representaciones específicas de una técnica de diseño (por ejemplo, para vuelta atrás árbol de búsqueda).
- Objetivo. Distinguimos tres objetivos destacados: un ejemplo de entrada/salida, una ejecución y una explicación teórica.

Además, para poder recoger y clasificar las representaciones de las distintas técnicas de diseño, ha sido conveniente introducir algunos criterios ad hoc:

- Completitud. Una ilustración puede aparecer completa o parcial (sólo una parte). También para las técnicas de vuelta atrás y programación dinámica encontramos ilustraciones simplificadas o que reflejan una definición inductiva en divide y vencerás.
- Subcategorías de las representaciones propias de cada técnica de diseño. Para divide y vencerás, se analizó el flujo de ejecución mostrado en las representaciones de la recursividad. Para vuelta atrás, se distinguió entre árbol de búsqueda potencial y generado. Para programación dinámica, se distinguió entre tablas que almacenan valores óptimos y tablas que almacenan las decisiones asociadas.

## 2.2.2 Resultado del análisis preliminar

A continuación se presentan algunos resultados del análisis preliminar realizado y presentado en [25] y [26].

Para el análisis preliminar, como se ha comentado anteriormente, los problemas se clasificaron en cuatro categorías: problemas de juegos (por ejemplo, n reñas), problemas de grafos (por ejemplo, problema del viajante), otros problemas de decisión (por ejemplo, dos subconjuntos de igual suma) y otros problemas de optimización (por ejemplo, mochila 0/1).

Se distinguió entre las siguientes representaciones gráficas: árbol de búsqueda potencial y generado, dependientes del dominio y propias de la informática. A su vez, los árboles de búsqueda pueden estar representados de forma completa, simplificada o parcial.

Los hallazgos principales fueron:

- Las representaciones gráficas usadas mayoritariamente son: los árboles de búsqueda (55%) (desglosados en dos tercios de árboles potenciales y un tercio de árboles generados), las dependientes del dominio (30%) y propias de la informática (grafos, 15,46%).
- Los árboles de búsqueda potenciales sólo están completos o simplificados. Los generados pueden ser de las tres clases, siendo parciales y completos los más usados (50% y 41,67%).

También se estudió el tratamiento dado a cada categoría de problema. Se obtuvieron los siguientes hallazgos:

- Los problemas con mayor número absoluto de ilustraciones son los de juegos (38,34%).
- Los problemas con mayor número relativo de ilustraciones son los de juegos (3,25 ilustraciones por libro) y grafos (3).
- Los árboles de búsqueda se usan para toda clase de problemas, usándose de forma exclusiva para otros problemas de decisión y problemas de optimización. Los problemas de juegos o de grafos se apoyan más en representaciones dependientes del dominio.

También se analizó el objetivo de las ilustraciones. Se encontró lo siguiente:

- El uso de las ilustraciones es variado: explicación (46,67%), ejecución (35%) o entrada/salida (18,33%).
- Las ejecuciones y las explicaciones se usan con toda clase de problemas, mientras que la entrada/salida se limita a problemas de juegos o de grafos (26,83% de las representaciones de estas clases de problemas).
- Los árboles de búsqueda sólo se usan en ejecuciones (81,71% de éstas) o explicaciones (53,57%). Las distintas clases de árboles difieren en su objetivo. Los árboles parciales sólo se usan para ejecuciones. Los árboles potenciales completos suelen usarse (62,5%) para ejecuciones. Los árboles potenciales simplificados suelen usarse para explicaciones (81,82%).
- En dos casos de juegos se ha encontrado una secuencia de estados del tablero para ilustrar la ejecución del algoritmo, junto con su árbol potencial y generado.
- Las representaciones dependientes del dominio se usan para todos los fines, aunque principalmente para explicaciones (47,06%, por ejemplo, para mostrar la generación o validez de candidatos). Sin embargo, las ilustraciones de grafos no se usan para ejecuciones.





### 3. Categorías de catalogación

Con el objeto de poder homogeneizar el análisis para las tres técnicas de diseño y tras varias reuniones y puestas en común de resultados se han establecido las siguientes categorías:

- Numeración. Para cada técnica de diseño se incluye la numeración de las figuras recogidas en su respectivo informe técnico.
- Libro. Indica la obra de la que se ha extraído la figura.
- Clase de algoritmo. Indica la técnica de diseño a la que pertenece la imagen.
- Clase de problema. Nuestra aproximación ha sido pragmática, identificando clases de problemas que agrupan de forma “razonable” los problemas encontrados para cada técnica de diseño. Se han reagrupado los tipos de problemas en función del problema que resuelven, homogeneizando así los tipos de problemas incluidos en las tres técnicas de diseño .
  - Problema. Permite identificar, dentro de la clase de problema, el problema concreto al que pertenece la imagen.
- Objetivo. Distinguimos cuatro objetivos bien destacados. Una imagen que forma parte del enunciado del problema, una imagen que forma parte del diseño del algoritmo, una imagen que forma parte de la ejecución o un ejemplo de análisis del algoritmo.
- Subobjetivo. Para cada objetivo se distinguen una serie de subobjetivos dependiendo de la finalidad de la imagen. Así, por ejemplo, para el objetivo Enunciado del problema, podemos encontrar distintos subobjetivos: un ejemplo de definición del problema, un ejemplo para definir una propiedad auxiliar, un ejemplo de entrada, un ejemplo de salida válida o inválida, una salida parcial, varias salidas posibles, todas las salidas posibles o la salida óptima.

Cuando el objetivo es mostrar diseño del algoritmo o ejecución tenemos los siguientes subobjetivos:

  - En DyV: partición, Combinación, Part+Comb, Part+Rec, Rec+Comb y Part+Comb+Rec.

- Para Vuelta atrás: espacio de estados, generación de candidatos, organización por niveles, comprobación de validez de un candidato, solución parcial, solución parcial inválida, 1ª solución completa válida, todas las soluciones válidas o la solución óptima.
- Programación dinámica. Formato de tabla, dependencia entre celdas, ecuaciones recursivas.
- Genericidad. Una imagen puede estar representada de forma genérica, concreta, simplificada o semigenérica.
- Representación del estado. El estado se puede representar de distintas formas, por ejemplo con: Vector, vector+índices, tabla, matriz, tablero, pila de control, árbol n-ario, árbol binario, árbol sintáctico, grafo, aritmética, simple, índices, diagrama geométrico, fórmula química o suprimido, en este caso no se representa el estado del algoritmo con ninguna información.
- Completitud del estado. El estado puede estar representado de forma completa, parcial o simplificada.
- Representación del tiempo. El tiempo se representa mediante 1, 2 o una secuencia de estados, con solapamiento, con una secuencia de estado más solapamiento, con un árbol de recursión, grafo de dependencia, árbol de búsqueda, una explicación mediante texto otra representación con historia distinta de las anteriores o puede no estar representado con ningún tipo de imagen, por lo que ponemos indefinido o N/A.
- Completitud del tiempo. El tiempo puede estar representado de forma completa, parcial o abreviada.
- Atributos de representación del tiempo. En este caso, los árboles de recursión pueden ser ascendentes o descendentes o ambos. Los árboles de búsqueda se pueden clasificar en potenciales o generados. Indiferente, Valores, decisiones o ambas.

Una vez establecidas las categorías y con el objeto de llevar a cabo el análisis de las figuras se ha creado una tabla Excel con una columna por cada una de las categorías indicadas anteriormente y para cada imagen se recopila la siguiente información:

Nº	Libro	Problema	Técnica	Clase de problema	Objetivo	Subobj.	Repr. estado	Complet estado	Repr. Tiempo	Complet. tiempo	Atrib. repr.
----	-------	----------	---------	-------------------	----------	---------	--------------	----------------	--------------	-----------------	--------------

Tabla 4. Tabla Excel – Categorías de catalogación definitivas

Con el objeto de aclarar las categorías de catalogación, en la siguiente sección se muestran las categorías más relevantes con ejemplos ilustrativos.

### 3.1 Objetivo de representación

En esta categoría se recopilan todas las figuras de acuerdo a su objetivo de representación. Tras analizar las figuras recopiladas para la técnica de vuelta atrás distinguimos 3 objetivos de representación bien destacados:

- Enunciado: en este objetivo de representación se recogen todas las figuras que representan alguna definición, restricción o propiedad relacionada con el enunciado del algoritmo.
- Diseño: engloba todas las figuras que están relacionadas con el diseño del algoritmo.
- Ejecución: contiene las figuras que representan parte de la ejecución del algoritmo.

Así, para cada objetivo de representación, se distinguen distintos subobjetivos de representación (véase Tabla 5) atendiendo a la información incluida en cada una de las figuras.

En la Tabla 5 se muestran los objetivos y subobjetivos de representación diferenciados para la técnica de vuelta atrás.

Enunciado	Diseño	Ejecución
Definición	Comprobación validez	1ª solución completa válida
Entrada	Espacio de estados	Solución parcial inválida
Entrada representación	Generación de candidatos	Todas las soluciones válidas
Salida	Organización de niveles	Solución óptima
Salida válida		
Salida inválida		
Salida(parte)		

Tabla 5. Subobjetivos de representación por objetivos de representación

A continuación se muestran distintos ejemplos ilustrativos sobre cada objetivo y subobjetivo de representación.

La Figura 3 muestra una ilustración para el algoritmo del salto de caballo. Consiste en, dada una posición inicial, representada en la imagen por el carácter “c”, en un tablero de ajedrez recorre todas las casillas del tablero únicamente con los movimientos del caballo sin repetir ninguna casilla. La figura muestra, de forma sombreada, las casillas con los posibles movimientos que puede realizar el caballo. En este caso, la ilustración se utiliza para definir parte del enunciado del problema, en concreto, los posibles movimientos que puede realizar el caballo dentro del tablero (objetivo: enunciado, subobjetivo: definición).

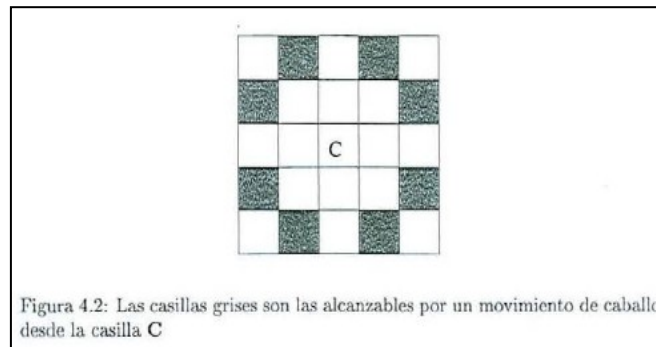


Figura 3. Tablero movimientos del caballo de ajedrez

En la Figura 4 se puede observar en la ilustración superior un ejemplo de entrada (objetivo) para el problema del coloreado de grafos. En la figura inferior se muestra el grafo asociado al grafo planar que se utiliza para representar la la entrada del problema y en el que se puede ver que no puede haber dos vértices adyacentes con el mismo color. El subobjetivo de representación en este caso es “entrada representación”.

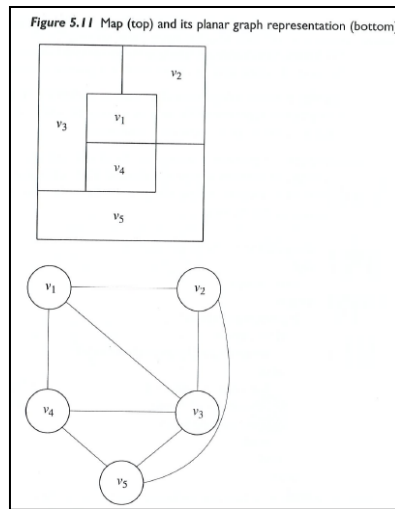


Figura 4. Grafo planar-grafo

La Figura 5 muestra una ilustración para el algoritmo de las 8-reinas. Se trata de un tablero de ajedrez donde se tienen que colocar las ocho reinas sin que se amenacen entre sí. En la ilustración se ve un ejemplo de salida válida (subobjetivo) para el problema. Se incluye dentro del objetivo de representación enunciado, porque representa el cumplimiento de las propiedades del problema. Es decir, la salida es válida debido a que se han colocado las ocho reinas en el tablero sin haber dos reinas en la misma fila, columna o diagonal.

column	1	2	3	4	5	6	7	8
row 1				Q				
2						Q		
3								Q
4		Q						
5							Q	
6	Q							
7			Q					
8					Q			

Figure 7.1 One solution to the 8-queens problem

Figura 5. Tablero de ajedrez 8-reinas

En la Figura 6 se puede ver una ilustración para el algoritmo de las 8-reinas. En concreto, se representa un ejemplo de salida inválida para el problema ya que, no se pueden colocar dos reinas en la misma diagonal. El objetivo de la representación es mostrar que no se cumplen las restricciones indicadas en el enunciado del problema. Por ello, se incluye dentro del objetivo de representación enunciado y su subobjetivo

es representar una salida inválida.

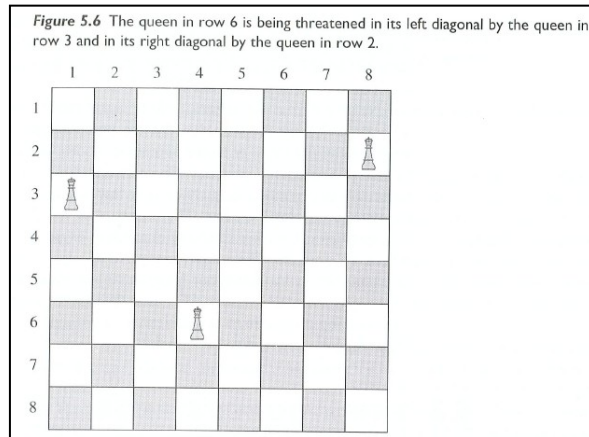


Figura 6. Tablero de ajedrez 8-reinas

En la Figura 7 se representa una ilustración para el algoritmo del dominó. En concreto, la figura muestra parte de la salida correspondiente a una partida del dominó. El objetivo de representación de esta figura es ilustrar cómo se deben colocar las fichas para cumplir con el enunciado del problema. En este caso el subobjetivo de representación es mostrar parte de la salida.

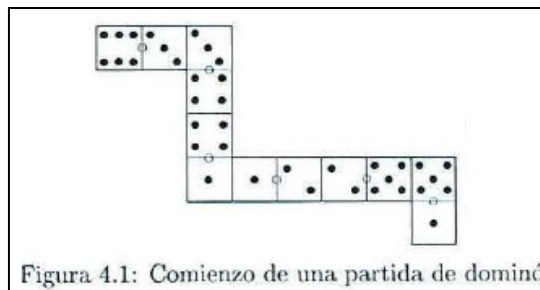


Figura 7. Ejemplo de partida del dominó

La Figura 8 muestra una ilustración para el algoritmo de las 8-reinas. En ella se puede ver un ejemplo de cómo se van a enumerar las diagonales del tablero para poder comprobar la validez de la solución (subobjetivo). Es decir, si cuando se va a posicionar una reina en el tablero se cumplen las restricciones del problema. Por ello el objetivo de representación de esta imagen es el diseño del algoritmo.

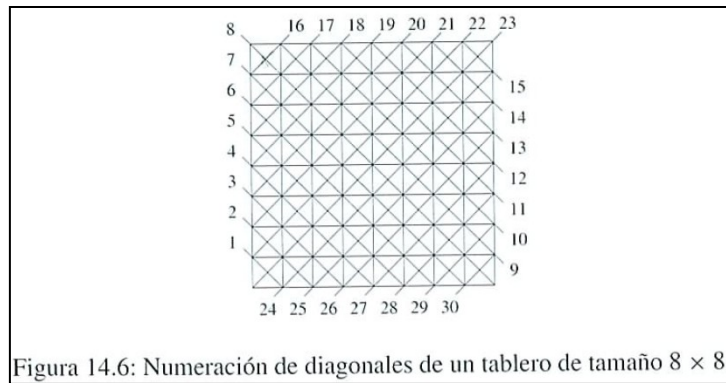


Figura 8. Tablero de ajedrez - comprobación de validez

En la Figura 9 se muestra una ilustración que representa parte del espacio de estados para el problema de las 4-reinas. En cada nodo del árbol puede observarse la reina que se va a colocar en cada momento, la fila y la columna donde se posiciona. Forma parte del diseño del algoritmo porque representa cómo se construye y recorre el espacio de estados para el problema planteado. El subobjetivo de representación de esta imagen es representar el espacio de estados.

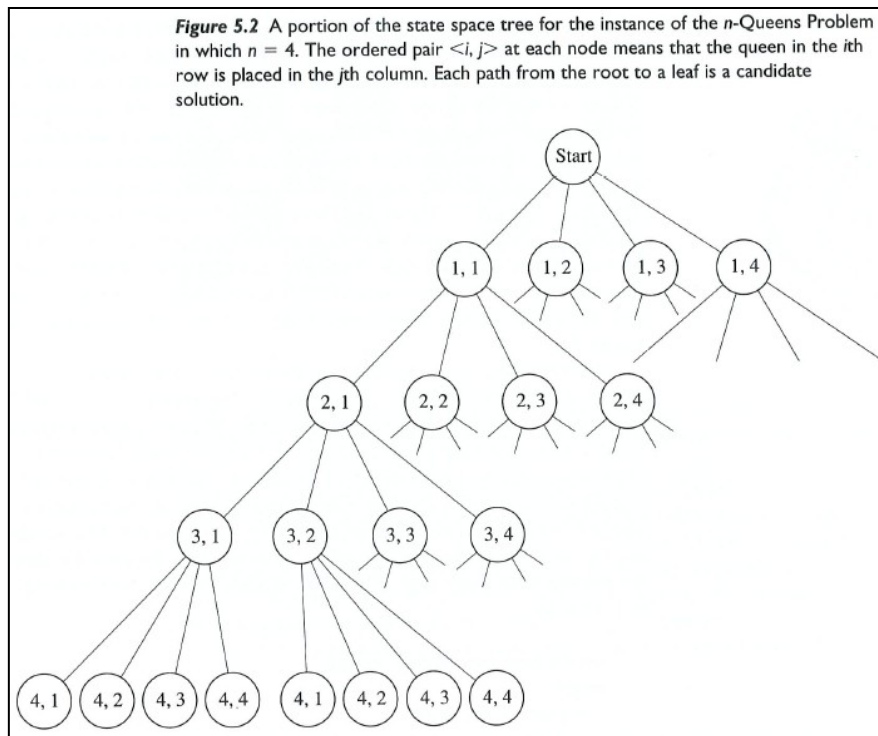


Figura 9. Árbol de búsqueda potencial

La Figura 10 ilustra los movimientos permitidos para el caballo en el tablero. En este caso, muestra como se numeran los 8 tipos de movimientos que puede realizar el

caballo en el tablero partiendo de la fila y columna  $\langle i,j \rangle$ . El objetivo de representación es diseño ya que, indica cómo se van generando los candidatos en cada etapa del problema (subobjetivo de representación). Cada movimiento permitido es un candidato posible.

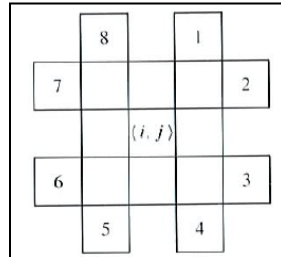


Figura 10. Tablero parcial con movimientos del caballo válidos

La Figura 11 muestra cómo se organizan los niveles y cómo se numeran las casillas para el problema del cuadrado latino. Por ello, su objetivo de representación es diseño.

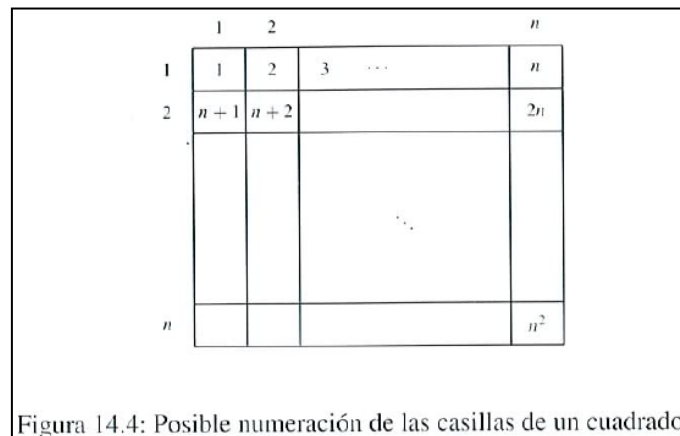


Figura 14.4: Posible numeración de las casillas de un cuadrado.

Figura 11. Numeración por niveles – cuadrado latino

En la Figura 12 se representa, para el problema de las 4-reinas, cómo se recorre el espacio de estados a medida que se va ejecutando el algoritmo. Los nodos marcados con una “x” indican que no continúa por ese camino porque no se cumplen las restricciones del problema. En este caso la ejecución termina cuando encuentra la primera solución válida (subobjetivo de representación). En este caso, para el análisis, esta ilustración se ha partido en dos ilustraciones simples. El objetivo de representación es mostrar la ejecución del algoritmo.



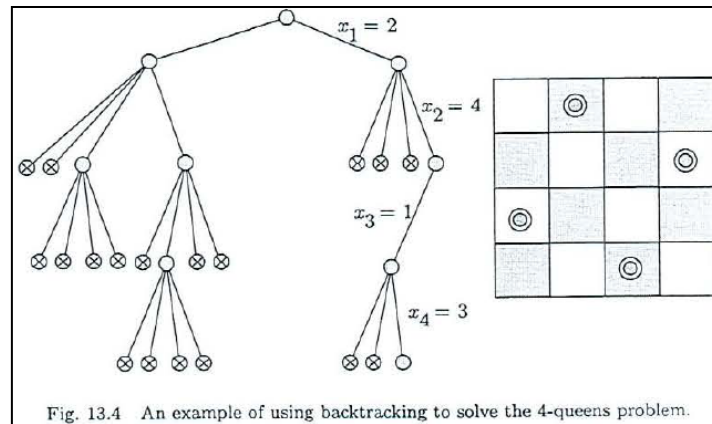


Figura 12. Árbol de búsqueda generado – tablero solución

El objetivo de representación de la Figura 13 es ejecución porque muestra dos estados de la ejecución del algoritmo. Es un ejemplo de solución parcial inválida (subobjetivo), porque no cumple con las restricciones del problema, en este caso las reinas se amenazan por estar en la misma columna (primera figura) o en la misma diagonal (segunda figura).

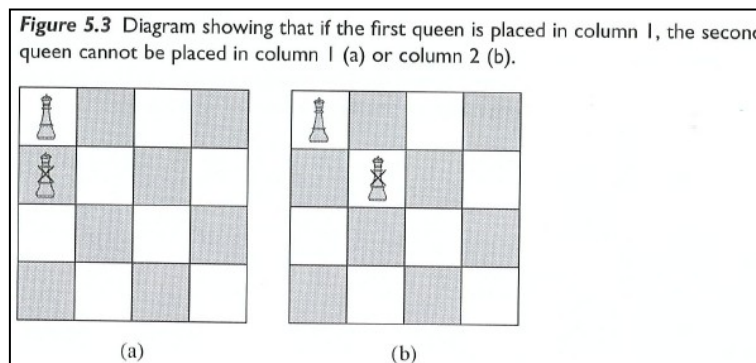


Figura 13. Tablero 4-reinas – 2 estados

El objetivo de representación de la Figura 14 es ejecución, ya que, muestra cómo se va recorriendo el espacio de estados y las podas que va realizando para el problema de la suma de subconjuntos. En la figura se muestran los nodos que no pueden llevar a un resultado válido con una x. El subobjetivo de la figura es mostrar todas las soluciones válidas.

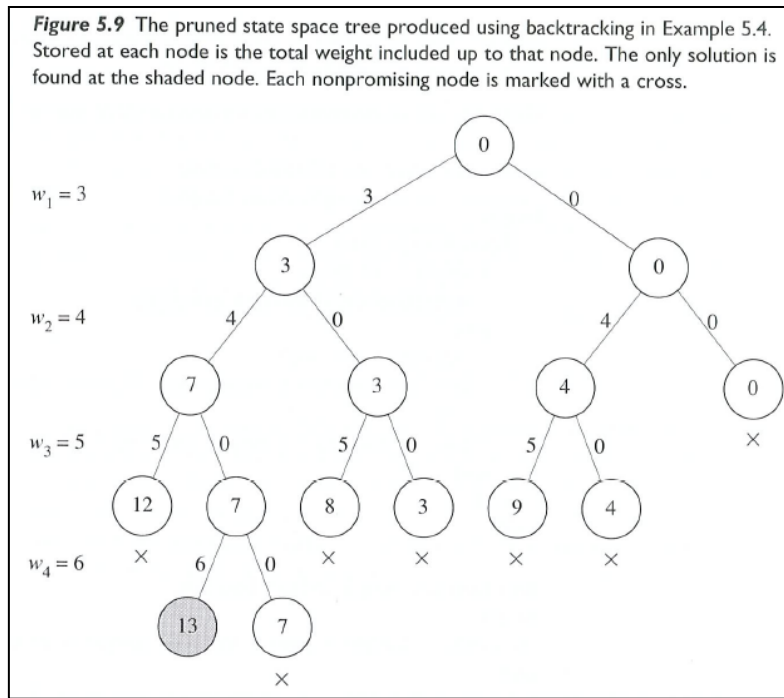


Figura 14. Árbol de búsqueda generado

En la Figura 15 se muestra como va recorriendo el espacio de estados mientras se ejecuta el algoritmo para el problema de la mochila 0/1. Los nodos que no pueden llevar a una solución mejor que la encontrada hasta el momento los poda y los marca con una x. Además, en cada nodo se almacena el beneficio total de los artículos introducidos en la mochila hasta el momento, el peso total, el beneficio que se podría obtener con los objetos restantes. El objetivo de representación en este caso es mostrar la ejecución del algoritmo y el subobjetivo es mostrar la solución optima, porque el algoritmo está destinado a encontrar la solución óptima para el problema planteado.

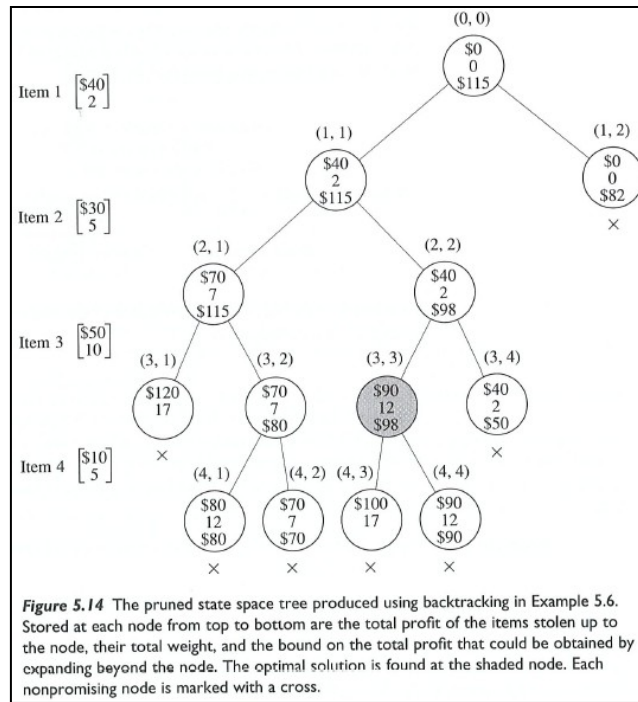


Figura 15. Árbol de búsqueda generado

### 3.2 Genericidad

Atendiendo a las figuras analizadas hemos observado que una imagen puede ser genérica: no incluye datos de entrada concretos para un algoritmo determinado, concreta: para unos datos de entrada concretos o semigenérica: para unos datos de entrada concretos pero no se especifican todos los datos.

La Figura 16 muestra una imagen genérica (genericidad) porque ilustra el espacio de estados válido para cualquier conjunto de datos de entrada para el problema del vendedor.

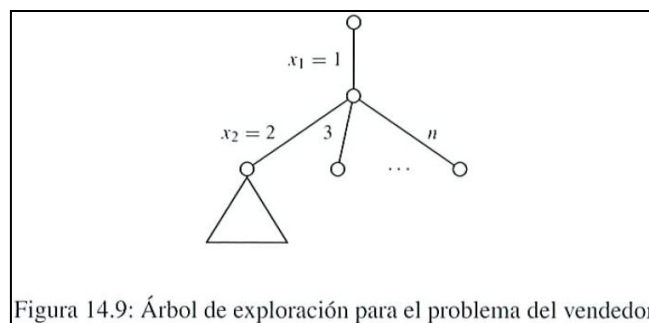


Figura 16. Árbol de búsqueda abreviado

En la Figura 12 se muestra una ilustración del espacio de estados para el problema de las 4-reinas. Por ello, esta imagen se considera concreta.

La Figura 17 muestra el espacio de estados para el problema del coloreado de grafos para un grafo de 3 vértices, pero no se incluye unos datos de entra específicos por ello, se considera semigenérica.

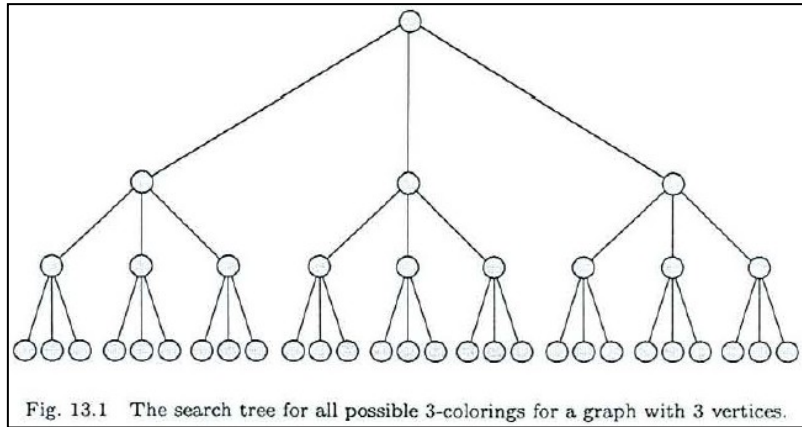


Figura 17. Árbol de búsqueda potencial

### 3.3 Representación y completitud del estado

En esta categoría se recogen las figuras que representan el estado del algoritmo en un instante de tiempo determinado. Así, el estado puede estar representado de forma completa, parcial o simplificada. Se considera que el estado está simplificado cuando se representa en parte la figura pero se da a entender de algún modo que continua pero no se representa (ver Figura 12). La completitud del estado se considera parcial, cuando se representa parte de la información (ver Figura 7) .

En general, el estado del algoritmo se puede representar mediante distintas estructuras, por ejemplo: con un vector con o sin índices, un diagrama geométrico, una matriz, un tablero, una tabla, un grafo, puede estar representado por otras estructuras o no estar representado con ninguna información, en este caso se considera que el estado está suprimido.

En concreto, para la técnica de vuelta atrás, el estado se puede estar representado

por: un tablero, índices, un grafo, un dato simple, puede estar suprimido o estar representado por otras estructuras.

Con el objeto de aclarar cómo se representa el estado, a continuación se muestran distintos ejemplos ilustrativos.

La Figura 12 muestra una ilustración que se divide en dos ilustraciones simples, la de la izquierda, el árbol de búsqueda, cuya representación del estado está suprimida, es decir, no muestra ninguna información relacionada con el estado del algoritmo en un instante de tiempo determinado, y donde la completitud del estado se puede observar que está simplificada porque, no se muestra el árbol de búsqueda de forma completa, pero se muestran las aristas a modo de simplificación. La ilustración de la derecha representa una salida válida con las cuatro reinas colocadas. En este caso el estado del algoritmo se representa de forma completa (aparece el tablero completo) mediante un tablero.

En la Figura 9 se puede observar un árbol de búsqueda donde el estado está representado mediante índices que indican la reina que se está colocando y la fila y columna del tablero donde se coloca dicha reina. La completitud del estado es simplificada porque no se representa la información de todos los nodos del árbol de búsqueda.

La Figura 18 ilustra un ejemplo donde el estado esta representado mediante un grafo completo. En este caso, como el objetivo de ambas ilustraciones es el mismo, para el análisis se ha considerado como una única figura.

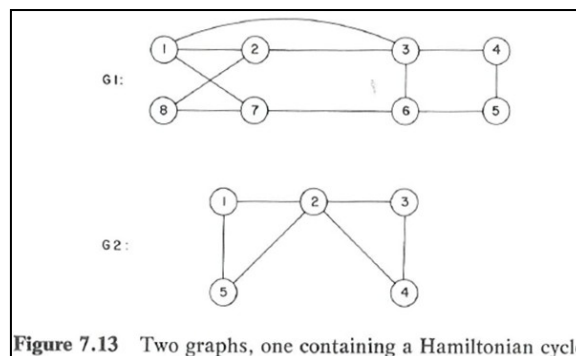


Figure 7.13 Two graphs, one containing a Hamiltonian cycle

Figura 18. Grafo completo

Se ha considerado que el estado es un dato simple cuando se representa mediante algún dato de entrada del problema como por ejemplo, un número entero que representa el peso de un objeto. En la Figura 15 se ilustra un ejemplo donde el estado se representa mediante un dato simple.

Otra forma de representar el estado es mediante otras estructuras, como por ejemplo un mapa como se muestra en la Figura 4. La ilustración superior mostrada en la parte superior donde se muestra una representación planar de un grafo.

### **3.4 Representación y completitud del tiempo**

En general, en esta categoría se recogen las figuras que representan el tratamiento del tiempo de ejecución del algoritmo. Así, el tiempo puede estar representado de forma completa (se representa toda la figura completa), parcial (por ejemplo para los árboles de búsqueda, se ilustran algunas de sus ramas ver Figura 19) o abreviada (por ejemplo, para los árboles de búsqueda se incluyen algunas de sus ramas pero se utilizan símbolos para representar la continuidad de la ilustración, ver Figura 16) . También se recogen datos sobre los atributos que se utilizan para cada una de las representaciones.

Para la técnica de vuelta atrás, el tiempo puede estar representado por: un árbol de búsqueda, una secuencia de estados + solapamiento (en este caso se almacena distinta información sobre la historia de la ejecución del algoritmo ver Figura 20), 2 estados (ver Figura 13) o puede estar indefinido o dependiendo del tipo de ilustración no aplicar almacenar la representación del tiempo. Por ejemplo, si el objetivo de representación es enunciar una propiedad del algoritmo o es el diseño del algoritmo y la ilustración es un tablero, no tiene sentido representar el información sobre el tratamiento del tiempo de ejecución del algoritmo (véase Figura 6).

En el caso de los árboles de búsqueda, pueden estar ilustrados de forma completa, parcial o abreviada y puede ser generado (cuando el árbol de búsqueda se genera en tiempo de ejecución) o potencial (cuando se representa en la fase de diseño del

algoritmo). Estos datos (potencial y generado) se almacenan como atributos de representación.

Atendiendo a todas las categorías definidas anteriormente tenemos que, la Figura 19 muestra un árbol de búsqueda para el problema de la mochila, cuyo objetivo de representación es ejecución (no continúa por las ramas que no pueden llegar a una solución óptima), tiene como subobjetivo de representación encontrar la solución óptima, esta representada para un ejemplo concreto (genericidad), el estado está representado mediante índices que indican el orden en el que se va recorriendo el espacio de estados, no se muestran todos los nodos con los índices por lo que el estado está simplificado (completitud) y el tiempo se representa mediante un árbol de búsqueda parcial (se representa parte de sus ramas) generado (se genera en tiempo de ejecución).

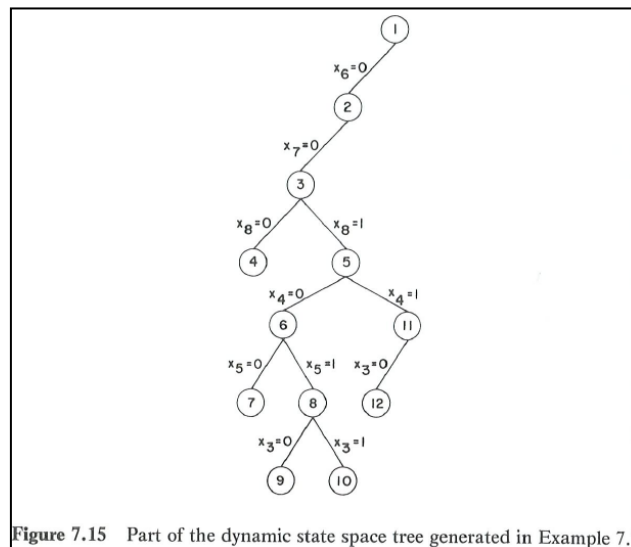


Figure 7.15 Part of the dynamic state space tree generated in Example 7.7

Figura 19. Árbol parcial generado

La Figura 16 ilustra un árbol de búsqueda para el problema de los ciclos hamiltonianos, su objetivo de representación es el diseño, porque ilustra el espacio de estados (subobjetivo) para el problema indicado, es genérica ya que no se representa para unos datos de entrada concretos, no se incluye información sobre el estado del algoritmo en un instante de tiempo determinado (el estado está suprimido) y la completitud del estado está simplificada (no se muestran todos los nodos pero si se indica con puntos suspensivos la continuidad), el tiempo se representa mediante un

árbol de búsqueda abreviado potencial. Es abreviado porque muestra parte del árbol, con subárboles comprimidos en triángulo o suprimidos, y arcos suprimidos con puntos suspensivos.

En la Figura 20 se puede observar una ilustración para el problema de las 4-reinas donde se muestra la ejecución del algoritmo. El algoritmo para cuando encuentra la primera solución válida. El estado está representado mediante un tablero completo y el tiempo se representa con la secuencia de tableros, donde cada tablero almacena la secuencia de movimientos que va realizando para colocar las reinas (por eso se considera que hay solapamiento). Los puntos de la imagen representan estados inválidos que no cumplen con las restricciones del problema.

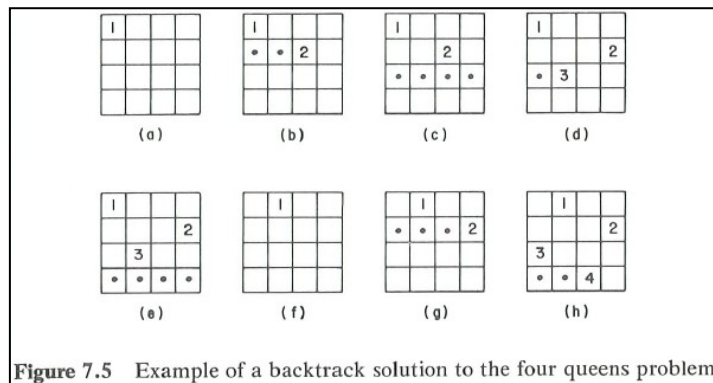


Figure 7.5 Example of a backtrack solution to the four queens problem

Figura 20. Tableros – secuencia + solapamiento

La Figura 13 muestra dos estados de la ejecución del algoritmo para el problema de las 4-reinas. En este caso el tiempo esta representado por dos estados, que representan una solución parcial inválida.



## 4. Análisis de figuras

Como se ha comentado en el subcapítulo 2.1, se han consultado 14 obras bibliográficas de las cuales 6 no presentan ninguna figura relacionada con la técnica de vuelta atrás. Se han recopilado un total de 25 tipos de problemas de los cuales en 6 no se presenta ninguna ilustración.

En la Tabla 1 se puede observar el porcentaje de figuras por categorías de problemas.

### Categorías de catalogación

Atendiendo a las categorías de catalogación definidas en el capítulo 3 obtenemos los resultados que se muestran a continuación.

En la Tabla 6 se puede ver que las representaciones mayoritarias corresponden a representaciones para datos de entrada concretos (67,74%) sobre un problema determinado.

Total	Concreta	Genérica	Semigenérica
62(100%)	42(67,74%)	11(17,74%)	9(14,52%)

Tabla 6. Porcentajes de figuras por genericidad

La Tabla 7 muestra el porcentaje de figuras según el objetivo de representación. Puede observarse que las representaciones en su mayoría se utilizan para ilustrar el diseño del algoritmo (42,94%) y alguna propiedad del enunciado del mismo (32,26%).

Total	Enunciado	Diseño	Ejecución
62(100%)	20(32,26%)	26(42,94%)	16(25,81%)

Tabla 7. Porcentajes de figuras por objetivo de representación

De acuerdo con la Tabla 8 las representaciones más utilizadas corresponden a aquellas cuyo subobjetivo de representación es ilustrar el espacio de estados o la 1ª primera solución completa válida.

1ª solución completa válida	8(12,90%)
Comprobación validez	1(1,61%)
Definición	4(6,45%)
Entrada	7(11,29%)
Entrada representac.	1(1,61%)
Espacio de estados	21(33,87%)
Generación candidatos	2(3,23%)
Organización de niveles	1(1,61%)
Salida	1(1,61%)
Salida (parte)	2(3,23%)
Salida inválida	2(3,23%)
Salida válida	3(4,84%)
Solución óptima	3(4,84%)
Solución parcial	1(1,61%)
Solución parcial inválida	1(1,61%)
Todas soluciones válidas	4(6,45%)
Total	62(100%)

Tabla 8. Porcentajes de figuras por subobjetivo de representación

Como se puede observar en la Tabla 9 la principal forma de representar el estado de ejecución en un instante de tiempo determinado es mediante tableros (30,65%), en el 24,19% de las ilustraciones no se representa el estado, y el 17,74% utilizan índices para representarlo.

Total	Tablero	Índices	Grafo	Otros	Simple	Suprimido
62(100%)	19(30,65%)	11(17,74%)	9(14,52%)	1(1,61%)	7(11,29%)	15(24,19%)

Tabla 9. Porcentajes de figuras por representación del estado

La Tabla 10 indica que el estado se representa de forma completa (50%) o simplificada (45,77%).

Total	Completa	Simplificada	Parcial
62(100%)	31(50%)	29(45,77%)	2(3,23%)

Tabla 10. Porcentajes de figuras por completitud del estado

En la Tabla 11 se puede ver que el tratamiento del tiempo de ejecución del algoritmo, en el caso de estar representado en la figura, se suele representar por árboles de búsqueda (53,23%).

Total	Indefinido o N/A	Árbol búsq.	Secuencia+solapamiento	2 estados
62(100%)	26(41,94%)	33(53,23%)	2(3,23%)	1(1,61%)

Tabla 11. Porcentajes de figuras por representación del tiempo

Así, como se muestra en la Tabla 12 cuando se representa el tiempo de ejecución del algoritmo mediante árboles de búsqueda, estos se suelen incorporar de forma completa (63,64%) o de forma abreviada (33,33%).

Total	Completa	Abreviado	Parcial
33(100%)	21(63,64%)	11(33,33%)	1(3,03%)

Tabla 12. Porcentajes de figuras por completitud del tiempo para árboles de búsqueda

Los árboles de búsqueda más utilizados son los potenciales (63,34%). Esto tiene sentido porque este tipo de árboles se utilizan en la fase de diseño y ayudan a entender cómo diseñar el algoritmo y cómo se construye y recorre el espacio de estados.

Total	Generado	Potencial
33(100%)	12(36,36%)	21(63,34%)

Tabla 13. Porcentajes de figuras por atributos de representación

### Representaciones gráficas

En la Tabla 14 se muestra las diferencias en el uso de figuras entre los 14 autores. Se puede ver la diferencia del número de imágenes incluidas por los autores, así, hay 6 autores sin ilustración alguna (42,86%), 5 autores que ofrecen menos de 8 ilustraciones (35,71%) y 3 autores que incluyen más de 10 ilustraciones (21,43%), Horowitz y Sahni [7], Neapolitan y Naimipour [14] y Martí Oliet et al. [13]).

Solamente 1 autor [13](7,14%) incluye alguna figura en las cinco categorías de problemas. Puede observarse que en la categoría de problemas de juegos se incluye un mayor número de ilustraciones. Esto se debe a, para este tipo de problemas, que se comprende mejor el algoritmo con ilustraciones gráficas con las que se pueden representar los posibles movimientos.

Clase de problema	Total	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
Juegos (J)	25(40,32%)	4			3	4		7	5					2	
Grafos (G)	18(29,03%)	4					3	2	5				2	2	
Cadenas(C)	1(1,61%)							1							
Decisión (D)	10(16,13%)	3						4	3						
Optimización (O)	8(12,90%)	2	1					3	1					1	
TOTAL	62(100%)	13	1		3	7		17	14				3	4	

Tabla 14. Porcentajes de figuras por clase de problemas y autores

En la Tabla 15 se puede ver que las representaciones más utilizadas son con datos de entrada concretos (70%). Se puede observar también que las representaciones para datos de entrada concretos son más frecuentes en problemas de juegos (35%) y grafos (23,33%).

Respecto a las representaciones genéricas no parece haber una diferencia significativa entre las distintas categorías de problemas.

Clase de problema	Total	Concreta	Genérica	Semigenérica
Juegos (J)	25(40,32%)	21	4	
Grafos (G)	18(29,03%)	14	1	3
Cadenas(C )	1(1,61%)			1
Decisión (D)	10(16,13%)	3	4	3
Optimización (O)	8(12,90%)	4	2	2
TOTAL	62	42	11	9

Tabla 15. Porcentajes de figuras por clase de problemas y genericidad

La Tabla 16 indica que en los problemas de juegos el estado se representa mayoritariamente mediante tableros (31,67%). Así, los tableros sólo se utilizan en problemas de Juegos. También se puede ver que el estado también se suele representar mediante índices (18,33%) en 4 de las 5 categorías de problemas, sin haber entre ellas una diferencia significativa.

Solamente en problemas de decisión y optimización se representa el estado mediante datos simples. Esto tiene sentido, porque en este tipo de problemas se utilizan datos numéricos para representar distinta información (por ejemplo, para el problema de decisión de conjuntos de igual suma, en cada estado del algoritmo se va actualizando la suma de los elementos que se llevan seleccionados hasta el momento).

Clase de problema	Total	Tablero	Índices	Grafo	Otros	Simple	Suprimido
Juegos (J)	25(40,32%)	19	4				2
Grafos (G)	18(29,03%)		3	9	1		5
Cadenas(C )	1(1,61%)						1
Decisión (D)	10(16,13%)		2			3	5
Optimización (O)	8(12,90%)		2			4	2
TOTAL	62	19	11	9	1	7	15

Tabla 16. Porcentajes de figuras por clase de problemas y representación del estado

Respecto a la representación del tiempo de ejecución, en la Tabla 17 se puede ver que el tiempo está indefinido mayoritariamente en problemas de juegos (26,67%) y de grafos (16,67%). Lo que no sorprende teniendo en cuenta que las representaciones

mayoritarias en los problemas de juegos son tableros utilizados para ilustrar enunciado del problema (véase Tabla 28) y este tipo de figuras no almacenan información sobre el tiempo de ejecución del algoritmo (véase Tabla 18).

Los árboles de búsqueda se utilizan para representar el tiempo en las cuatro categorías de problemas, estando más presentes en problemas de decisión y de optimización con mayor número relativo de ilustraciones (100% en ambos casos).

La Tabla 17 también indica que las representaciones compuestas (con dos estados) o las representaciones con secuencias de estados + solapamientos se dan en problemas de juegos, lo que no sorprende al centrar su objetivo en probar distintos movimientos posibles por parte del jugador.

Clase de problema	Total	Indefinido o N/A	Árbol búsq.	Secuencia+solapamiento	2 estados
Juegos (J)	25(40,32%)	16	6	2	1
Grafos (G)	18(29,03%)	10	8		
Cadenas(C)	1(1,61%)		1		
Decisión (D)	10(16,13%)		10		
Optimización (O)	8(12,90%)		8		
<b>TOTAL</b>	<b>62</b>	<b>26</b>	<b>33</b>	<b>2</b>	

Tabla 17. Porcentajes de figuras por clase de problemas y representación del tiempo

Atendiendo a la representación del tiempo de ejecución del algoritmo respecto a la representación del estado (véase Tabla 18) se obtiene que el tiempo suele estar indefinido para tableros (26,67%) y grafos (15%).

El estado en los árboles de búsqueda aparece mayoritariamente suprimido (25%) o representado mediante índices (18,33%).

Repr. espacio	Total	Indefinido o N/A	Árbol búsqueda	Secuencia+solapamiento	2 estados
Tablero	19	16		2	1
Índices	11		11		
Grafo	9	9			
Otros	1	1			
Simple	7		7		
Suprimido	15		15		
<b>TOTAL</b>	<b>62</b>	<b>26</b>	<b>33</b>	<b>2</b>	<b>1</b>

Tabla 18. Porcentajes de figuras por representación del espacio y representación del tiempo

Tabla 19 muestra el análisis de los tipos de árboles de búsqueda por autor. Puede observarse que se representan mayoritariamente árboles de búsqueda potenciales.

Sólo 3 [7][11][17] autores representan los dos tipos de árboles de búsqueda. El autor [10] no incluye no incluye ningún tipo de árbol de búsqueda.

Atributo de representación	Total	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
Generado	12(36,36%)	4	1			2			4					1	
Potencial	21(63.64%)	5				1		10	3			2			
TOTAL	33(100%)	9	1			3		10	7			2		1	

Tabla 19. Porcentajes de figuras por atributos de representación y autor

La Tabla 20 muestra un análisis de las representaciones halladas según el tipo de árbol de búsqueda y su completitud en el tiempo. Hay una preferencia general por los árboles completos (63,64%) y mientras que los árboles abreviados se utilizan en su mayoría para representar árboles de tipo potencial.

Atributo de representación	Total	Completo	Abreviado	Parcial
Generado	12(36,36%)	10	1	1
Potencial	21(63.64%)	11	10	
TOTAL	33	21	11	1

Tabla 20. Porcentajes de figuras por atributos de representación y completitud del tiempo

Atendiendo a las distintas representaciones utilizadas por los autores para representar los árboles abreviados, éstos se han clasificado en tres tipos: árboles con subárboles suprimidos (SS, véase Figura 9), árboles con subárboles comprimidos en triángulo o suprimidos, y arcos suprimidos con puntos suspensivos (CTAPS, véase Figura 16), y con subárboles comprimidos en triángulo (ST, véase Figura 1).

Como se puede ver en la Tabla 21 la principal representación de árboles de búsqueda potenciales abreviados son árboles que contienen subárboles comprimidos en triángulo o suprimidos, y arcos suprimidos con puntos suspensivos (72,73%).

Atributo de representación	Total	SS	CTAPS	ST
Generado	1	1		
Potencial	10	1	8	1
TOTAL	11	2	8	1

Tabla 21. Tipos de árbol abreviados

Si nos fijamos en la Tabla 22, donde se muestra el análisis de la completitud del tiempo por autor para árboles de búsqueda, parece haber un sesgo claro en el caso de Martí

Oliet et al [13] con un claro uso de árboles de búsqueda abreviados (generalmente de tipo CATPS, véase Figura 16 y ST véase Figura 1). Cabe mencionar que sólo dos autores representan árboles abreviados.

Complejidad del tiempo	Total	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
Completo	21(63,64%)	8	1			3		1	5			2	1		
Abreviado	11(33,33%)							9	2						
Parcial	1(3,03%)	1													
TOTAL	33(100%)	9	1			3		10	7			2	1		

Tabla 22. Porcentajes de figuras por completitud del tiempo para árboles de búsqueda y autores

### Objetivo de Representación

La Tabla 23 muestra el análisis de las representaciones halladas según su objetivo de representación y clase de problemas. Puede observarse que el uso principal de las representaciones es ilustrar el diseño del algoritmo (41,94%, siendo su uso independiente de la clase de problemas) o mostrar datos de entrada o propiedades que se deben cumplir en el enunciado del problema (32,26%). El uso de figuras para ilustrar el enunciado del problema se limita a problemas de juegos o grafos.

Objetivo de rep.	Total	Juegos (J)	Grafos (G)	Cadenas (C)	Decisión (D)	Optimización (O)
Enunciado	20(32,26%)	10	10			
Diseño	26(41,94%)	8	5	1	8	4
Ejecución	16(25,81%)	7	3		2	4
TOTAL	62	25	18	1	10	8

Tabla 23. Porcentajes de figuras por objetivo de representación y clase de problemas

La Tabla 24 muestra el subobjetivo de las representaciones por objetivo de representación. En ella se puede ver que los subobjetivos de representación, cuando el objetivo de la ilustración es mostrar el enunciado del problema, son mayoritariamente representar datos de entrada (12,90%) o mostrar la salida del algoritmo (12,90%).

Respecto a las representaciones cuyo objetivo es mostrar el diseño del algoritmo, su subobjetivo mayoritario es ilustrar el espacio de estados. Lo que tiene sentido porque el espacio de estados permite definir o diseñar el proceso de resolución de un problema.

Cuando el objetivo de representación es mostrar la ejecución del algoritmo, su subobjetivo se centra principalmente en encontrar la 1ª solución completa válida

(12,90%).

Subobjetivo de representación	Total	Enunciado	Diseño	Ejecución
1ª solución completa válida	8(12,90%)			8
Comprobación validez	1(1,61%)		1	
Definición	4(6,45%)	4		
Entrada	7(11,29%)	7		
Entrada representación	1(1,61%)	1		
Espacio de estados	21(33,87%)		21	
Generación candidatos	2(3,23%)		2	
Organización de niveles	1(1,61%)		1	
Salida	1(1,61%)	1		
Salida (parte)	2(3,23%)	2		
Salida inválida	2(3,23%)	2		
Salida válida	3(4,84%)	3		
Solución óptima	3(4,84%)			3
Solución parcial	1(1,61%)		1	
Solución parcial inválida	1(1,61%)			1
Todas soluciones válidas	4(6,45%)			4
<b>TOTAL</b>	<b>62(100%)</b>			

Tabla 24. Porcentajes de figuras por subobjetivo de representación y clase de problemas

La Tabla 25 muestra que los autores usan las figuras con fines variados. Solamente parece haber un sesgo claro en el caso de Martí Oliet et al [13] con un uso claro de ilustraciones para explicar el diseño del algoritmo. Sólo 3 autores ([7][11][14]) ofrecen representaciones para los tres objetivos de representación.

Objetivo de repr.	Total	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
Enunciado	20(32,26%)	3			3	2		3	5			1	3		
Diseño	26(41,94%)	5				2		14	3			2			
Ejecución	16(25,81%)	5	1			3			6				1		
<b>TOTAL</b>	<b>62(100%)</b>	<b>13</b>	<b>1</b>		<b>3</b>	<b>7</b>		<b>17</b>	<b>14</b>			<b>3</b>	<b>4</b>		

Tabla 25. Porcentajes de figuras por objetivo de representación y autores

En la Tabla 26 se puede ver que los subobjetivos de representación utilizados por los autores en las figuras son variados. Cabe destacar los casos de Martí Oliet et al [13] con un uso claro de ilustraciones para representar el espacio de estados, hecho que era de esperar porque centra su objetivo de representación en el diseño del algoritmo. Horowitz y Sahni [7] y Neapolitan y Naimipour [14], con un uso mayoritario de las figuras también para ilustrar el espacio de estados. De hecho, todas las figuras que incluyen estos últimos autores, cuyo objetivo de representación es mostrar el diseño del algoritmo, se centran en representar el espacio de estados.



Subobjetivo.de representación	Total	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
1ª solución completa válida	8(12,90%)	2				3			3						
Comprobación validez	1(1,61%)							1							
Definición	4(6,45%)	1			1			1	1						
Entrada	7(11,29%)	1				1			2			1	2		
Entrada representación	1(1,61%)								1						
Espacio de estados	21(33,87%)	5				1		10	3			2			
Generación candidatos	2(3,23%)							2							
Organización de niveles	1(1,61%)							1							
Salida	1(1,61%)												1		
Salida (parte)	2(3,23%)				1			1							
Salida inválida	2(3,23%)					1			1						
Salida válida	3(4,84%)	1			1			1							
Solución óptima	3(4,84%)	2							1						
Solución parcial	1(1,61%)					1									
Solución parcial inválida	1(1,61%)								1						
Todas soluciones válidas	4(6,45%)	1	1						1					1	
TOTAL	62(100%)	13	1		3	7		17	14			3	4		

Tabla 26. Porcentajes de figuras por subobjetivo de representación y autores

En la Tabla 27 se puede ver un análisis de la genericidad de las ilustraciones según su objetivo de representación. Puede verse que las representaciones para datos de entrada concretos se suelen utilizar mayoritariamente para ilustrar el enunciado o la ejecución del algoritmo. Mientras que las ilustraciones con datos de entrada genéricos o semigenéricos se usan para representar el diseño del algoritmo.

Objetivo de representación	Total	Concreta	Genérica	Semigenérica
Enunciado	20(32,26%)	20		
Diseño	26(41,94%)	6	11	9
Ejecución	16(25,81%)	16		
TOTAL	62	42	11	9

Tabla 27. Porcentajes de figuras por subobjetivo de representación y genericidad

En la Tabla 28 se muestra el análisis de las representaciones halladas según la representación del estado y su objetivo de representación. En ella se puede ver que los tableros se utilizan principalmente (52,63%) como parte del enunciado del problema. Los grafos se usan exclusivamente como parte del enunciado del problema. Los índices y los datos simples se utilizan tanto para ilustrar el diseño del algoritmo como para ilustrar la ejecución sin haber una diferencia significativa entre ellos.

Objetivo de representación	Total	Tablero	Índices	Grafo	Otros	Simple	Suprimido
Enunciado	20(32,26%)	10		9	1		
Diseño	26(41,94%)	5	6			2	13
Ejecución	16(25,81%)	4	5			5	2
<b>TOTAL</b>	<b>62</b>	<b>19</b>	<b>11</b>	<b>9</b>	<b>1</b>	<b>7</b>	<b>15</b>

Tabla 28. Porcentajes de figuras por objetivo de representación y representación del estado

Centrándonos en la representación del estado mediante tableros o grafos, en la Tabla 29 podemos ver que los tableros destinados a mostrar parte del enunciado del algoritmo se utilizan mayoritariamente (80%) para representar algún estado de la salida, parte de la salida, una salida inválida o la salida válida del mismo (véase la Tabla 24 para ver qué subobjetivos tienen como objetivo mostrar el enunciado del problema).

Por otro lado, teniendo en cuenta que los grafos tienen como objetivo el enunciado del algoritmo, tiene sentido que su subobjetivo de representación sea mostrar una definición o representar los datos de entrada del algoritmo.

Subobjetivo.de representación	Tablero	Grafo
1ª solución completa válida	3	
Comprobación validez	1	
Definición	1	3
Entrada	1	5
Entrada representación		1
Espacio de estados		
Generación candidatos	2	
Organización de niveles	1	
Salida	1	
Salida (parte)	2	
Salida inválida	2	
Salida válida	3	
Solución óptima		
Solución parcial	1	
Solución parcial inválida	1	
Todas soluciones válidas		
<b>TOTAL</b>	<b>19</b>	<b>9</b>

Tabla 29. Figuras por subobjetivo de representación y representación del estado

En la Tabla 30 se puede ver el análisis de la representación del estado según su objetivo de representación. Puede verse que los árboles de búsqueda se suelen utilizar para mostrar el diseño o la ejecución del algoritmo. Así, la secuencia de estados más solapamiento se utiliza exclusivamente para mostrar la ejecución del algoritmo.

Objetivo de representación	Total	Indefinido o N/A	Árbol de búsqueda	Secuencia+solapamiento	2 estados
Enunciado	20(32,26%)	20			
Diseño	26(41,94%)	5	21		
Ejecución	16(25,81%)	1	12	2	1
<b>TOTAL</b>	<b>62</b>	<b>26</b>	<b>33</b>	<b>2</b>	<b>1</b>

Tabla 30. Porcentaje de figuras por objetivo de representación y representación del tiempo

De la Tabla 31 se extrae que los árboles de búsqueda potenciales se utilizan para ilustrar el diseño del algoritmo mientras que los generados para mostrar la ejecución.

Objetivo de representación	Total	Generado	Potencial
Enunciado	20(32,26%)		
Diseño	26(41,94%)		21
Ejecución	16(25,81%)	12	
<b>TOTAL</b>	<b>62</b>	<b>12</b>	<b>21</b>

Tabla 31. Porcentaje de figuras por objetivo y atributos de representación

### Formato Gráfico

A continuación se ofrece la distribución de características de las figuras en cuanto a elementos de formato, ordenado por categoría de problemas. Los elementos más destacados son etiquetas, distinción de nodos especiales y clases de nodos.

Con el objeto de poder realizar un análisis más en detalle, el mismo se ha desglosado en varias tablas. En la Tabla 32 puede verse que hay gran variación en el uso de formatos, aunque destacan los nodos con vértices circulares o redondeados (30 figuras), las etiquetas en las ramas del árbol de búsqueda (23 figuras) y etiquetas dentro de cada nodo (19 figuras). Generalmente, las etiquetas en la rama corresponden a decisiones con identificador y valor de variable. También es interesante la distinción de nodos especiales (por ejemplo, representación de nodos inválidos o nodos solución).

Formato	Total	J	G	C	D	O
Etiquetas dentro nodo	19	5	3		5	6
Etiquetas fuera nodo	7	2	2		1	2
Etiquetas en todas las ramas	23	3	4	1	9	6
Etiquetas en algunas ramas	4	1	1		1	1
Etiquetas nivel	5		2		2	1
Distinción nodos inválidos	7	3	2		1	1
Distinción nodos solución	6	1	3		1	1
Nodo rectangular	2				1	1
Nodo circular	30	6	7	1	9	7
Nodo punto	1		1			
TOTAL		21	25	2	30	26
Figuras		6	8	1	10	8

Tabla 32. Porcentaje de formatos en árboles y clases de problemas

Tras observar las distintas ilustraciones, se han analizado las diferentes etiquetas que se pueden encontrar en los árboles de búsqueda tanto dentro como fuera de cada nodo. Así, con el objeto de poder determinar la tendencia de representación simbólica para nodos inválidos, nodos válidos y otros datos que aparecen en las figuras se va a analizar la frecuencia de aparición de la siguiente información: cruz (X), sombreado (S), número de orden (Nº) y otros datos dependientes del problema (DP, por ejemplo, coordenadas donde se sitúan las reinas, número de color seleccionado para colorear el grafo, la suma acumulada para los subconjuntos, etc.). También se va a analizar si la etiqueta está fuera del nodo en la parte inferior (INF) o en un lateral (LAT), si los nodos están numerados con caracteres, bien en profundidad, bien por niveles (NºL) y si el nodo está marcado por un círculo (PR). Por último, se va a tener en cuenta si se representa otra información en el nodo (OTR), por ejemplo, si el nodo inválido está marcado con la letra B.

En la Tabla 33 puede observarse que el elemento más usado son las etiquetas situadas dentro del nodo y dependientes del dominio.

Formato	X	S	Nº	DP	INF	LAT	Nº L	PR	OTR	TOTAL
Etiquetas dentro nodo	2	5	5	9			2	1		24
Etiquetas fuera nodo	4		1	1	5	2			1	14
TOTAL	6	5	6	10	5	2	2	1	1	38

Tabla 33. Porcentajes tipo etiquetas nodos de árboles y representación simbólica

En la Tabla 34 se puede ver que la etiqueta más utilizada para indicar un nodo inválido es la “x” y los nodos solución se marcan de forma sombreada “s”.

Formato	X	S	Otros	Total
Nodos inválidos	6		1	
Nodos solución		5		
TOTAL	6	5	1	12

Tabla 34. Porcentajes distinción nodos y representación simbólica

Además de los árboles de búsqueda, en la bibliografía se encuentran ilustraciones de tableros y grafos.

Los tableros se utilizan en problemas de juegos. En la Tabla 35 puede verse la variación en el uso de formatos para la representación de los distintos elementos del tablero. No parece haber ningún sesgo especial.

Formato	J
Símbolo Dama	3
Símbolo Reina	4
Punto	1
Letras	3
Números	7
Colores	2
Líneas	1
Etiquetas fuera del tablero	3
TOTAL	24
Figuras	19

Tabla 35. Porcentajes de formatos en tableros y categoría juegos (J)

En la Tabla 36 puede observarse la variación en el uso de formatos para la representación de los distintos elementos del grafo. Se puede ver que en 7 de las 9 figuras se utilizan nodos circulares de los cuales 5 se encuentran numerados con números.

Formato	G
Nodo circular	7
Nodo punto	2
Numeración nodo letra	1
Numeración nodo número	5
Numeración nodo letras y subíndices numéricos	2
Etiquetas en aristas	1
TOTAL	19
Figuras	9

Tabla 36. Porcentajes de formatos en tableros y categoría grafos (G)

En la Tabla 37 se puede ver el uso de formatos, para árboles de búsqueda, entre los distintos autores. Parece haber una preferencia clara de Horowitz y Sahni [7] por las etiquetas dentro de nodos y en todas las ramas y de Martí Oliet et al. [13] por las etiquetas en todas las ramas y nodos circulares.

Formato	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	Total
Etiquetas dentro nodo	7	1			2		1	6			2				19
Etiquetas fuera nodo	2							4				1			7
Etiquetas en todas las ramas	6				1		10	4			2				23
Etiquetas en algunas ramas	3				1										4
Etiquetas nivel	1							4							5
Distinción nodos inválidos	1				2			4							7
Símbolo	B				x			x							
Distinción nodos solución					1			4				1			6
Sombreado					1			4							5
Nodo rectangular	1	1													2
Nodo circular	8				3		10	7			2				30
Nodo punto													1		1
TOTAL	29	2			11		30	39			6	3			
Figuras	9	1			3		10	7			2	1			33

Tabla 37. Porcentajes de formatos de árboles y autores

## 4.1 Resultado del análisis de figuras

Se han consultado 14 libros de texto, hallándose que 6 no incluían la técnica de vuelta atrás. Así, de los 25 tipos de problemas, categorizados en las 5 categorías de problemas (de juegos (J), grafos (G), cadenas (C), Decisión (D) y optimización (O)), 6 no presentan ninguna figura.

Atendiendo a la categorización de los problemas, obtenemos que los problemas con mayor número de representaciones son los de juegos (40,32%) seguidos de los de grafos (29,03%).

Se han analizado las distintas categorías. Se ha obtenido lo siguiente:

- Los autores representan mayoritariamente figuras para datos de entrada concretos (67,74%, dependientes del dominio. Por ejemplo, el tablero con las 4 reinas, dos subconjuntos de igual suma para un conjunto de objetos datos).
- El objetivo de uso de las figuras es variado: enunciado (32,26%), diseño

(41,94%) y ejecución (25,81%).

- Los subobjetivos se centran principalmente en ilustrar el espacio de estados (33,87%) o la primera solución completa válida (12,90%).
- Para representar el estado en un instante de tiempo determinado, los autores utilizan mayoritariamente tableros (30,65%) o no lo representan (24,29%), es decir, aparece suprimido. Atendiendo a su completitud tenemos que generalmente los autores no suelen representar el estado de forma parcial (3,23%).
- Respecto a la representación del tiempo, los autores suelen utilizar árboles de búsqueda para representarlo (53,23%). Esto tiene sentido porque de este modo representan el modo en el que se va recorriendo el espacio de estados en tiempo de ejecución. Generalmente, los árboles de búsqueda los representan de forma completa (63,64%) o abreviada (33,33%, completos pero simplificados).

Se ha estudiado el tratamiento dado a cada categoría de problemas frente al estado y al tiempo de representación y se han encontrado los siguientes hallazgos:

- Respecto al estado de representación, para problemas de juegos, el tipo de representación más utilizada son los tableros. De hecho, los tableros sólo se utilizan en dicha categoría de problemas.

Además, en 4 de las 5 categorías de problemas se suelen utilizar los índices para representar el estado mediante representaciones gráficas de árboles de búsqueda.

En el caso de problemas de decisión o de optimización, el estado se suele representar mediante datos simples (30% en el caso de problemas de decisión y 50% en problemas de optimización) o índices (20% y 25% respectivamente).

- Respecto a la representación del tiempo. Los árboles de búsqueda se usan en toda clase de problemas, usándose de forma exclusiva para los problemas de cadenas, problemas de juegos y problemas de optimización.

Los problemas de juegos o grafos se apoyan más en representaciones dependientes del dominio como por ejemplo, en tableros o grafos.

Se ha distinguido también el tratamiento dado al estado frente al tiempo y se han extraído los siguientes hallazgos:

- Cuando el estado se representa mediante grafos, el tiempo de ejecución suele estar indefinido.
- En el caso de que se represente mediante tableros, en su mayoría el tiempo de ejecución no está representado (84,21%) o se representa con una secuencia de estados y solapamiento de otro tipo de información acerca de la ejecución del algoritmo (10,53%).
- Para los árboles de búsqueda la representación del espacio suele estar suprimida (45,45%) o representarse mediante índices (33,33%) o datos simples (21,21%).

Se han analizado las representaciones gráficas que ilustran árboles de búsqueda. Se ha encontrado lo siguiente:

- Se distinguen dos tipos de árboles de búsqueda potencial (63,64%) y generado (36,36%).
- Los árboles de búsqueda generados pueden aparecer completos (83,33%), abreviados (completos pero simplificados, 8,33%) o dibujados parcialmente (8,33%).
- Los árboles de búsqueda potenciales sólo están completos o abreviados.
- Sólo dos autores utilizan árboles abreviados, aunque no coinciden en la clasificación de árboles utilizados. Uno de ellos ([14]) utiliza árboles de tipo SS y el otro ([13]) de tipo (CTAPS y ST).

También se ha analizado el objetivo de representación frente a otras categorías y se ha obtenido lo siguiente:

- Las representaciones que explican el diseño del algoritmo (por ejemplo, comprobación de validez de los candidatos, representar el espacio de estados, generación de candidatos, organización de los niveles o mostrar una solución parcial) se usan con toda clase de problemas. Las que representan la ejecución del algoritmo (mostrar la solución, ya sea la primera solución, la solución



completa válida o inválida) se utilizan con toda clase de problemas excepto con los de cadenas. Mientras que las ilustraciones que representan distinta información sobre el enunciado del problema (mostrar una definición, entrada y salida del algoritmo) sólo se usan en problemas de juegos y grafos.

- Las ilustraciones cuyo objetivo es el diseño del algoritmo se representan para datos de entrada genéricos (42,31%) o semigenéricos (24,62%).
- El estado se representa mediante tableros principalmente en el enunciado del problema, aunque también se usan en diseño y ejecución. Sin embargo, los grafos únicamente se usan en el enunciado del problema.
- El tiempo se representa principalmente mediante árboles de búsqueda en el diseño del algoritmo (80,77%) o ejecución del mismo (75%). Las distintas clases de árboles difieren en su objetivo, los árboles potenciales se usan para el diseño y los generados para la ejecución del algoritmo.
- En dos casos, en la categoría de juegos, se ha encontrado un secuencia de estados del tablero, con solapamiento de información, para ilustrar la ejecución del algoritmo, junto con su árbol de búsqueda.

Respecto al formato gráfico de los árboles de búsqueda, obtenemos lo siguiente:

- Los elementos más destacados son etiquetas, formato de nodos y distinción de nodos especiales.
- Especialmente las etiquetas se incluyen dentro del nodo y generalmente representan elementos dependientes del domino.
- Los árboles mayoritariamente se representan con etiquetas en todas las ramas (61%). Éstas etiquetas suelen corresponder a decisiones con identificador y valor de variable.
- Sólo el 15% de los árboles de búsqueda incluyen etiquetas que identifican el nivel del árbol.
- En el 21% de los árboles de búsqueda se resaltan los nodos inválidos y el 18% los nodos solución. Generalmente los nodos inválidos se representan mediante el símbolo "x" o mediante sombreado del nodo.
- Los nodos de los árboles de búsqueda se representan en su mayoría con nodos

circulares (91%).

Además, se obtiene que:

- Hay problemas que se apoyan más y se comprenden mejor con ilustraciones gráficas (por ejemplo, problemas de juegos o de grafos). Es más común que en otros ilustrar su entrada/salida.
- Las representaciones se apoyan más en datos de entrada concretos sobre todo para aquellos tipos de problemas que se comprenden mejor con ilustraciones gráficas.

## 5. Conclusiones y trabajos futuros

### 5.1 Conclusiones

Se ha llevado a cabo un análisis de las representaciones presentes 14 libros de algoritmos. En concreto, en este Trabajo Fin de Máster, se han analizado un total de 62 ilustraciones correspondientes a la técnica de vuelta atrás.

Para poder realizar el análisis se ha elaborado un informe técnico [21] que recoge, además de las ilustraciones, el pseudocódigo utilizado por los autores para resolver el problema. Se ha recogido esta información para que el informe pueda servir como referencia a la hora de buscar o consultar problemas resueltos con esta técnica.

Como se ha comentado en el subcapítulo 1.1 este Trabajo Fin de Máster forma parte de un proyecto que se centra en analizar las visualizaciones propias de las técnicas de diseño: divide y vencerás, vuelta atrás y programación dinámica. Así, para poder analizar las ilustraciones ha sido necesario por un lado, categorizar los tipos de problemas encontrados en los libros de texto y por otro, establecer distintas categorías de catalogación que permitiesen englobar todas las características y propiedades que presentaban las imágenes. Estas categorizaciones, además, han facilitado la homogeneización del análisis de las tres técnicas de diseño y ha permitido identificar tendencias generales.

El análisis se ha realizado de forma iterativa, con un gran esfuerzo de análisis y de síntesis posterior. Como resultado, se han identificado las preferencias de los autores de libros de texto de algoritmia, lo que permitirá poder extraer las características que deben tener las visualizaciones docentes sobre la técnica de diseño de vuelta atrás.

Cabe destacar que parte del trabajo presentado en este Trabajo Fin de Máster ha sido presentado en el XIV Simposio Internacional de Informática Educativa (SIIE 2012) [25] y en la revista IEEE-RITA [26] .

## 5.2 Trabajos futuros

Una vez realizado el análisis de las figuras incluidas en los libros de texto para la técnica de vuelta atrás, se plantean los siguientes trabajos futuros:

- Proponer la realización de un problema, a los alumnos, que permita determinar cuáles de los elementos gráficos incluidos en las ilustraciones recopiladas para la técnica de vuelta atrás son más útiles para comprender el funcionamiento del algoritmo.

Una vez analizadas las imágenes para las tres técnicas de diseño se les planteo un problema, sobre la técnica de divide y vencerás, a los alumnos de dos grupos de la asignatura de Diseño y Análisis de Algoritmos del Grado en Ingeniería Informática. El objeto de la realización de este problema era poder determinar qué elementos gráficos de los incluidos en las imágenes contenidas en los libros de texto eran de mayor utilidad para el entendimiento del algoritmo.

Para plantear el problema que debían resolver los alumnos (véase Anexo I – Problema planteado a los alumnos), nos reunimos los 4 investigadores y establecimos los criterios que debía cumplir. En dicho problema se les daba el código que resuelve el algoritmo y los alumnos debían responder a dos apartados. En el primer apartado se les proporcionaba una serie de imágenes extraídas de los libros de texto y se les pedía que indicasen, de los elementos gráficos que contenía la imagen, los que les parecían más útiles y en el segundo apartado se les pedía que representasen con sus figuras el comportamiento del algoritmo proporcionado.

Los cuestionarios pasados a los alumnos están pendientes de ser analizados para extraer los resultados.

- Partiendo de los resultados del análisis, se plantea extraer las características que deben tener las visualizaciones docentes y formular los principios de diseño gráfico para el desarrollo de un sistema de visualización de algoritmos. Aunque algunos principios pueden intuirse a partir de las definiciones, deben tener una formulación de reglas de uso.

- Comprobar que las conclusiones obtenidas quedan confirmadas con otras ilustraciones. Para ello, se deberían seleccionar otros libros de texto con otras imágenes distintas.
- En base a los principios de diseño gráficos identificados, desarrollar un sistema de visualización de programas orientado a técnicas de diseño en general.



## Bibliografía

- [1] R. Lister, "The naughties in CEd research: A retrospective," SIGCSE Inroads, vol. 1, no. 1, pp. 22-24, Marzo 2010.
- [2] G. Roessling, y T. L. Naps, "A testbed for pedagogical requirements in algorithm visualizations," en Proc. 7th Annual Conference on Innovation and Technology in Computer Science education, ITiCSE'02. New York, NY: ACM Press, pp. 96-100.
- [3] S. Khuri, "Designing effective algorithm visualizations." en E. Sutinen, Ed. Proceedings of the First Program Visualization Workshop. Joensuu, Finlandia: University of Joensuu, 2000, pp. 1-12.
- [4] L. Stern, y L. Naish. "Visual representations for recursive algorithms," en Proc. 33th SIGCSE Technical Symposium on Science Education, SIGCSE 2002. New York, NY: ACM Press, pp. 196-200.
- [5] J. Á. Velázquez-Iturbide, A. Pérez-Carrasco, y J. Urquiza-Fuentes, "A design of automatic visualizations for divide-and-conquer algorithms," Electronic Notes in Theoretical Computer Science, no. 224, pp. 113-120, Enero 2009.
- [6] M. Agrawala, W. Li, y F. Berthouzoz, "Design principles for visual communication." Communications of the ACM, vol. 54, no. 4, pp. 60-69, Abril 2011.
- [7] E. Horowitz y S. Sahni: Fundamentals of Computer Algorithms. Pitman (1978)
- [8] G. Brassard y P. Bratley: Fundamentos de algoritmia. Prentice-Hall (1996)
- [9] I. Parberry: Problems on Algorithms. Prentice Hall (1995)
- [10] J. Gonzalo Arroyo y M. Rodríguez Artacho: Esquemas algorítmicos. Enfoque metodológico y problemas resueltos. UNED (2000)
- [11] M. H. Alsuwaiyel: Algorithms Design Techniques and Analysis. World Scientific (1999)
- [12] M. T. Goodrich y R. Tamassia: Data Structures and Algorithms in Java. John Wiley & Sons, 2ª ed. ( 2001)

- [13] N. Martí Oliet, Y. Ortega y J. A. Verdejo: Estructuras de datos y métodos algorítmicos: ejercicios resueltos. Pearson (2004)
- [14] R. Neapolitan y K. Naimipour: Foundations of Algorithms. Jones and Bartlett (1997)
- [15] R. Sedgewick: Algorithms in Java. Addison-Wesley (2002)
- [16] S Baase y A Van Gelder: Computer Algorithms: Introduction to Design and Analysis. Addison-Wesley Longman (2000)
- [17] S. Sahni: Data Structures, Algorithms, and Applications in Java. Silicon Press (2005)
- [18] S. Skiena: The Algorithm Design Manual. Springer-Verlag (1998)
- [19] S. S. Tseng R. C. T. Lee, R. C. Chang e y Y. T. Tsai: Introducción al diseño y análisis de algoritmos. McGraw-Hill (2005)
- [20] T. H. Cormen, C. E. Leiserson, R. L. Rivest y C. Stein: Introduction to Algorithms. The MIT Press, 2ª ed (2001)
- [21] N. Esteban-Sánchez, J.Á. Velázquez-Iturbide, “Revisión Bibliográfica de Problemas Resolubles por la Técnica de Vuelta Atrás” , Serie de Informes Técnicos DLSI-URJC, nº. 2012-04.
- [22] A. Pérez Carrasco, J. Á. Velázquez Iturbide, F. Almeida Martínez, “Revisión bibliográfica de la representación de problemas de la técnica «divide y vencerás»,” Serie de Informes Técnicos DLSI1-URJC, no. 2012-02.
- [23] B. Sáenz Rubio, J. Á. Velázquez Iturbide, “Revisión bibliográfica de algoritmos de programación dinámica,” Serie de Informes Técnicos DLSI1-URJC, no. 2012-04.
- [24] B. Glaser, y A. Strauss, The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine, 1967.
- [25] N. Esteban, A. Pérez, B. Sáenz, J.A. Velázquez, (2012). Towards the identification of graphical principles for visualizing algorithm design techniques. 2012 International Symposium on Computers in Education (SIIE), Francisco José García, Lluís Vicent, Miquel Ribó, August Climent, José Luis Sierra y Antonio Sarasa (eds.),



IEEE Computer Society Press, 5 págs. Print ISBN:978-1-4673-4743-3, INSPEC  
Accession Number: 13247352.

- [26] N. Esteban, A. Pérez, B. Sáenz, J. A. Velázquez, (2013). Un análisis de ilustraciones impresas de tres técnicas de diseño de algoritmos. IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, IEEE-RITA, 2013, Volumen 1, Número 2 Pags.97-104.



## Anexo I – Problema planteado a los alumnos

De forma resumida, podemos explicar el algoritmo de ordenación rápida de la forma siguiente. Sea un vector con más de una celda. Se toma un elemento del vector como referencia (p.ej. el elemento de la izquierda), llamado elemento pivote. Los elementos del vector se reparten en dos partes, una que contiene los elementos menores que el pivote y otra que contiene los elementos mayores que el pivote. Una vez determinadas ambas partes, el elemento pivote se coloca en medio y cada parte se ordena recursivamente. Como resultado, el vector queda ordenado completamente.

El método auxiliar de partición suele ser un algoritmo iterativo que va colocando los elementos menores que el pivote a la izquierda del vector y los mayores a la derecha. Para ello, va hallando pares de elementos descolocados (es decir, un elemento menor que el pivote situado a la derecha y un elemento mayor situado a la izquierda) y los intercambia. Al final, también intercambia el pivote con el elemento menor situado más a la derecha. De esta forma, el pivote queda colocado entre los elementos menores y mayores.

El algoritmo puede implementarse de varias formas. A continuación incluimos una posible implementación:

```
public static void quickSort (int[] v) {
    ordenarRápido (v, 0, v.length-1);
}

public static void ordenarRápido (int[] v, int inf, int sup) {
    if (inf<sup) {
        int med = partir (v, inf, sup);
        if (inf<med) ordenarRápido (v, inf, med-1);
        if (med<sup) ordenarRápido (v, med+1, sup);
    }
}

public static int partir (int[] v, int inf, int sup) {
    int p = v[inf];
    int i = inf+1;
    int j = sup;
    int temp;
    do {
        for (; v[i]<=p && i<sup; i++);
        for (; v[j] >p ; j--);
        if (i<j) {
            temp = v[i];

```

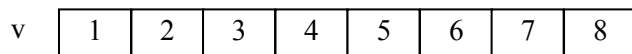
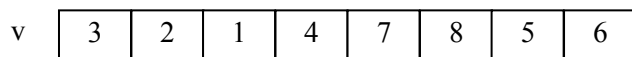
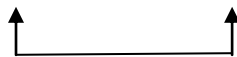
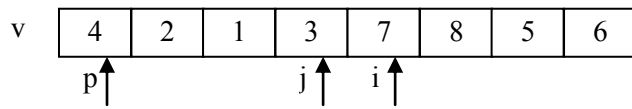
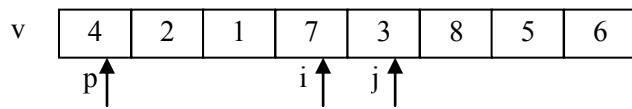
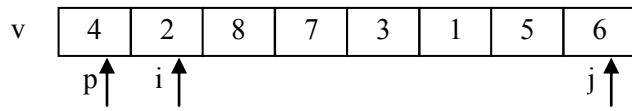
## Anexo I – Problema planteado a los alumnos

```
        v[i] = v[j];  
        v[j] = temp;  
    }  
} while (i<j);  
temp = v[inf];  
v[inf] = v[j];  
v[j] = temp;  
return j;  
}
```

Se pide:

1. Sea el siguiente vector de tamaño 8: {4,2,8,7,3,1,5,6}. Analice el comportamiento del algoritmo utilizando las figuras de las páginas siguientes. Se trata de siete figuras iguales, aunque con diferente estilo gráfico. Puntúe con un valor del 1 al 5 la utilidad de las figuras para comprender el comportamiento del algoritmo para este ejemplo, siendo: 1 – nada útil, y 5 – totalmente útil.
2. Explique con figuras dibujadas a mano el comportamiento del algoritmo para ordenar el vector {6,2,1,3,5,7,8,4}.

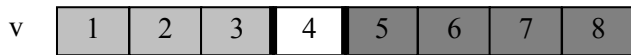
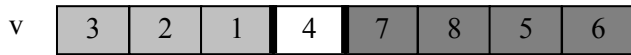
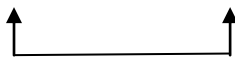
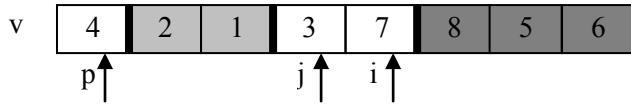
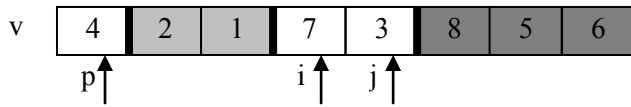
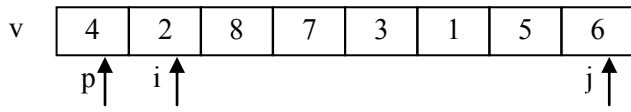
(Versión EF)







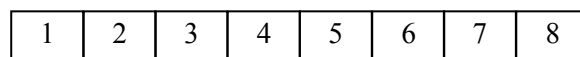
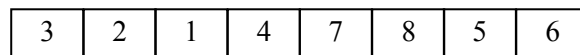
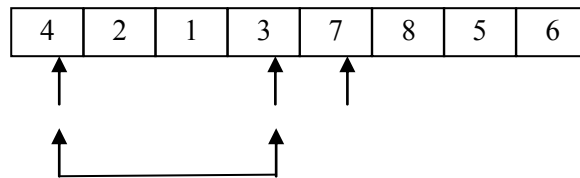
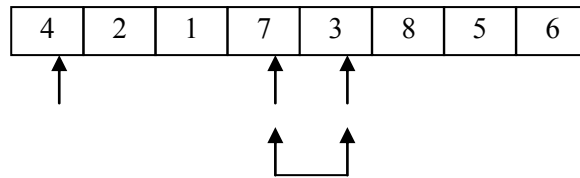
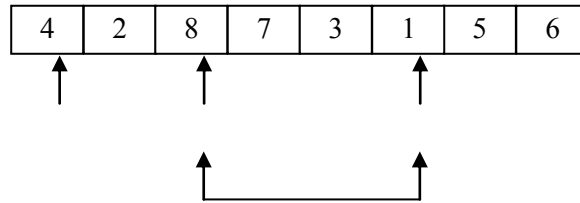
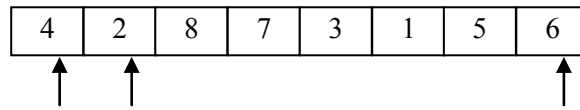
(Versión EFS)







(Versión F)



Anexo I – Problema planteado a los alumnos

(Versión S)

4	2	8	7	3	1	5	6
---	---	---	---	---	---	---	---

4	2	1	7	3	8	5	6
---	---	---	---	---	---	---	---

4	2	1	3	7	8	5	6
---	---	---	---	---	---	---	---

3	2	1	4	7	8	5	6
---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---